DEVELOPMENT OF THE BLUEBERRY IMPACT RECORDING DEVICE (BIRD) TO EVALUATE BLUEBERRY BRUISING IN THE MECHANICAL HARVEST PROCESS

by

PENGCHENG YU

(Under the Direction of Changying Li)

ABSTRACT

The overall goal of this study was to develop an instrumented sphere sensor, Blueberry Impact Record Device (BIRD), to quantitatively measure mechanical impacts endured by each blueberry fruit during the mechanical harvest process. The BIRD system has three components: the sensor (a one-inch sphere) with three single-axis accelerometers for data recording, the interface box to connect the sensor with the computer, and the custom-made computer software developed using LabVIEW to set up the BIRD sensor, download and process the data. The BIRD sensor has been fully calibrated with accuracy of -0.53~0.33% over the range of ±500 g and precision error of 0.63%. The BIRD sensor was successfully applied in the field to quantitatively describe what a typical blueberry experiences during the mechanical harvest process. Three major types of blueberry mechanical harvesters (rotary, slapper, sway) were evaluated using the BIRD sensor. The BIRD sensor data were also correlated with the actual bruising rate of blueberries in three cultivars. The BIRD sensor was proven to be an effective sensing device to evaluate blueberry bruising and the performance of mechanical harvesters. The information collected by the BIRD system can be used to improve blueberry mechanical harvesters with reduced blueberry bruising.

INDEX WORDS: Impact, Sensor, MEMS, Accelerometer, Microcontroller, LabVIEW, Instrumented sphere, Blueberry, Mechanical harvest, Bruising

DEVELOPMENT OF THE BLUEBERRY IMPACT RECORDING DEVICE (BIRD) TO EVALUATE BLUEBERRY BRUISING IN THE MECHANICAL HARVEST PROCESS

by

PENGCHENG YU

B.S., China Agricultural University, China, 2008

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2010

DEVELOPMENT OF THE BLUEBERRY IMPACT RECORDING DEVICE (BIRD) TO EVALUATE BLUEBERRY BRUISING IN THE MECHANICAL HARVEST PROCESS

by

PENGCHENG YU

Major Professor:	Changying Li
Committee:	Glen Rains
	Takoi Hamrita

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2010

ACKNOWLEDGEMENTS

Though only my name appears on the front page of this thesis, there are many more names that contributed much to its final production. I owe my gratitude to all those names that made this thesis possible. My deepest gratitude is to my advisor, Dr. Changying Li. I have been amazingly fortunate to have an advisor who allowed me this chance of graduate study and research with him, and a mentor who recovered my faltered steps and enlighten my mind. My research skills were nourished, my seeking of knowledge and truth were landed. I hope one day I would also become as good as him in my field of endeavor. My committee members, Dr. Glen Rains and Dr. Hamrita, have been always there listening and providing advice. I am equally grateful for their insightful comment and constructive points during different stages of my research. Mr. Timothy Rutland and Mr. John Gary Burnham are two individuals I am indebted to, for their unreserved efforts in making this research successful. I am also thankful to Mr. John Ed Smith and Mr.Robet D. Stanaland for their help with field tests. Dr. Peggy Ozias-Akins is also appreciated for providing the centrifuge for the sensor calibration. I am deeply grateful to the Triple R. Farm, the Allen Blueberry Farm and the DHL farm; growers behind those names showed their utmost generosity, they provided the harvesters, sacrificed their time and property to support this research. Still, I appreciate Dr. Fumiomi Takeda who has provided research data and valuable suggestions to support my research; Dr. Gerard Krewer’s support on this study is also appreciated. Most importantly, none of this has been possible without the love and patience of my family. Their love, concern, support and strength laid out each step towards where I am

now. I thank my deeply loved Yizhou, who accompanied with me through those difficult days and nights, and still fighting together with me to seek the dream of our life. Lastly, the USDA NIFA Specialty Crop Research Initiative is appreciated for providing funding support.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

The US produced 449.8 million lbs of blueberries in 2009 (Economic Research Service, USDA), accounting for 60% of the world production. Half of the blueberry production went to the fresh market and most of these fresh market blueberries are hand harvested (Strik and Yarborough 2005) . The increasing labor cost and low production efficiency prevent the blueberry industry from reaching a higher level. Therefore, it is crucial to improve the overall production efficiency of fresh market blueberries and the key factor is to advance the mechanical harvest technologies.

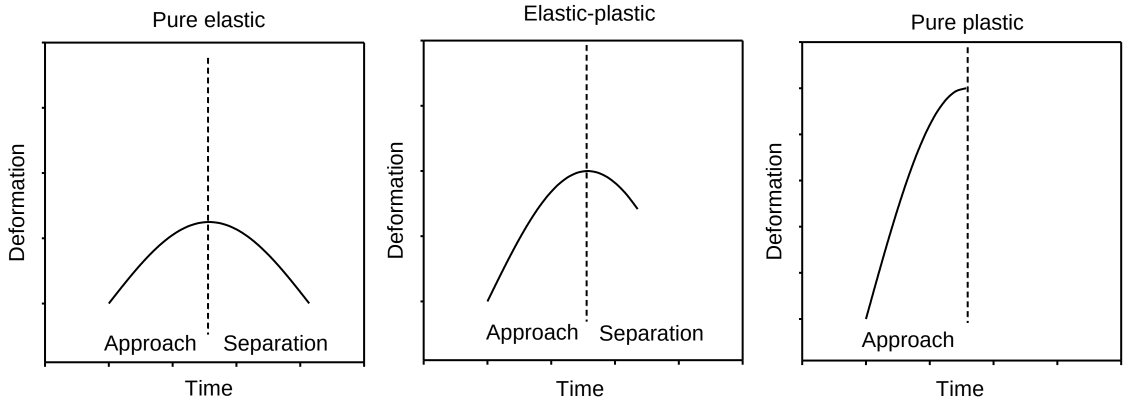
Use of commercial over-the-row mechanical harvesters improved labor productivity by nearly 60 times, cutting the cost by 85% (G.K. Brown 1996). However, there are two existing problems about the mechanical harvesters. The first is the excessive ground loss. In Brown's study, about 20% ~ 50% of the total blueberries that were shaken off by the three commercial harvesters was lost to ground. There were also ripe berries that were not removed from bushes. The second serious problem about the mechanical harvesters is that the majority(78%) of the mechanically harvested berries were bruised , which made the berries unmarketable to the fresh market (G.K. Brown 1996).

This thesis presents an effort to solve the second problem: mechanical bruising. Blueberries are exposed to frequent impact damages during the interactive process with a mechanical harvester. Damage can occur when the blueberry fruit collides with the interior surfaces or a mobile part of the harvester. However, there has been no previous knowledge about the quantity of impacts within the mechanical harvesters, and no effective way has been established to evaluate the harvester's potential damage to blueberries.

1.1 Mechanism of impact damage

Dynamic impacts can be defined as the case of rapid loading of forces created by collision being exerted and removed in a short period of time (Mohsenin 1980). Essentially, an impact between two contact bodies can be classified into four phases (Bowden and Tabor 1954):

- Initial elastic deformation during which the region of contact will be deformed elastically and will recover completely without residual deformation (Figure 1.1(a)).
- Onset plastic deformation during which the mean pressure exceeds the dynamic yield pressure of the material (fruit body) and the resulting deformation will not be fully recovered (Figure 1.1(b)).
- Full plastic deformation during which the deformation continues from elastic-plastic to fully plastic until the pressure falls below the dynamic yield pressure. (Figure 1.1(c))
- Elastic rebound during which a release of elastic stresses stored in both bodies takes place.



(a) (b) (c)

Figure 1.1 Deformation of the sphere body from three different types of impacts (Goldsmith 1960)

The fruit that undergoes pure elastic impact can recover completely without bruising damages. Depending on the actual impact forces, elastic-plastic impact can generate different degrees of bruising around the contact area. The deformation of the fruit can be partially recovered. In another case, pure plastic impact can smash the fruits without recovery from the dynamic deforming process. Higher deformation will generate more bruised area. To explain the two factors that affect the impact deformation of the fruit, a case study of comparison between blueberry's drop to hard surface (steel) and software surfaces is provided.

The combined deformation of an impact between an elastic sphere (fruit body) and a flat surface, as shown in figure 1.2, can be expressed as the following equation (Mohsenin 1980)

$$D = \left[\frac{15V^2 A m_1}{16\sqrt{R}} \right]^{\frac{2}{5}} \qquad (1)$$

As the total deformation D is the sum of d_1 (surface) and d_2 (sphere body), as illustrated in Figure 1.2; V is the initial relative velocity ($V = \sqrt{2gH}$, g=9.8 m/s^2); m_1 is the mass of the sphere; R is the radius of the sphere. A is defined in equation (2)

$$A = \frac{1-\mu_1^2}{E_1} + \frac{1-\mu_2^2}{E_2} \qquad (2)$$

μ and E are Poisson's ratio and Modulus of elasticity of the contacting bodies (Table 1.1).

Based on definition of Modulus of elasticity, the deformation of one contact body (d) can be calculated using equation (3).

$$d = \frac{F}{E \times S} \qquad (3)$$

E is the modulus of elasticity of the material and S is impact area size, F is the dynamic yield force generated between the two contact bodies. Because the sizes of contact areas of the two collided bodies’ are equal, according to equation (3), the relation of the deformations between two contact bodies is:

$$\frac{d_1}{d_2} = \frac{E_2}{E_1} \qquad (4)$$

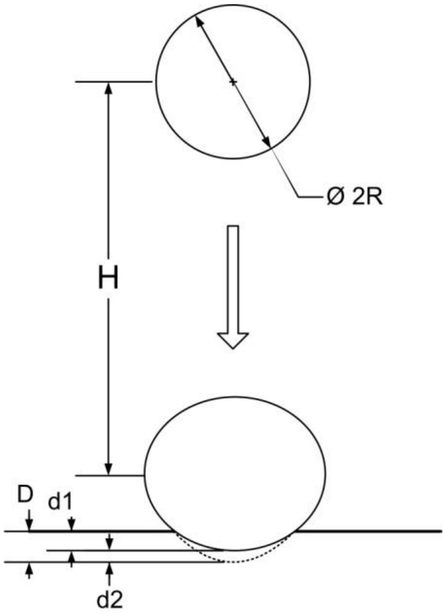


Figure 1.2 Impact between a sphere that dropped from H and a flat surface

Table 1.1 The Poisson’s ration and Modulus of elasticity of three materials; (Prussia, Tetteh et al. 2006; Chiabrando, Giacalone et al. 2009; Meththananda, Parker et al. 2009).

Materials		E(MPa)
Blueberry(sphere)	0.4	1.10
Urethane foam(flat surface)	0.1~0.4	0.27
Steel (flat surface)	0.27~0.3	2×10^5

According to equations (1), (2) and (4), the deformations of the berry fruits (m=1.5 g, d=2 cm) can be calculated under given drop conditions. We assume that the Poisson's ratio is 0.4 for the urethane foam, and is 0.3 for the steel. Based on the same drop height (H=0.1 m), the deformations of blueberries are 0.31 mm and 0.9 mm for the urethane foam and the steel respectively. This result provides an estimation of the fruit's deformation when dropped onto two different surfaces and confirms that a soft material can reduce the deformation of the fruit. The result of the calculation may have errors due to the approximation of the physical parameters of the contact materials.

On the other hand, for a fruit that drops on the same surface, increasing the drop height can result in a larger deformation of the fruit, based on equations (1), (2) and (4). So, reduction of the deformation (bruising) for single blueberry can be achieved by reducing the initial velocity (drop height), or change the contact surface to a soft material. This case study assumes that the impact is elastic. The actual bruising rate depends on the fruit's texture property under dynamic situations.

1.2 Using instrumented spheres to measure impact damage

"Pseudo fruits" have been employed to study the mechanical damage of various fruits and vegetables during the harvest and post harvest handling process (Bollen 2006). Essentially, a pseudo fruit is an instrumented sphere that can simulate actual fruits and record the forces generated due to those dynamic impacts. Products like PMS60 (Herold, Truppel et al. 1996), Impact Recording Device (IRD, Techmark MI), and the 'Electronic Potato' (PRT 100 and 200), (Canneyt, Tijskens et al. 2003) are three representative instrumented spheres. A detailed review of the applications of these instrumented spheres were presented in Chapters 2, 3, and 4.

However, all those sensors have limitations for the blueberry use. The first constraint is their size. For instance, the IRD has a minimum diameter of 57 mm (~ 2 inches) and weighs 96 grams. The PMS-60 has a diameter of 62 mm and weighs 180 g. The PTR has an elongated semi-ellipsoidal shape sensor with size of 53×53×83 mm. All these sensors are too big to go through the blueberry mechanical harvesters. The second constraint is their limitation of drop height: the IRD requires no more than 10 cm when dropped on a hard surface, and most of its applications have been focused on evaluations of packing lines rather than mechanical harvesters. The PMS-60 has a maximum drop height of 100 cm, while actual application for blueberries requires 150 cm or higher. The third constraint is the sampling speed. The PTR200's sampling frequency is only 100Hz, which would not be adequate to record the shape of the dynamic impact curves.

To our knowledge, no electronic sphere sensor has been developed and applied for small fruits like blueberries. The main challenge lies in the small size of the berry fruit (<30 mm). In order to measure impact damages of blueberries, it is necessary to design an instrumented sphere that has a much smaller size and a sufficient sensing range. The sensor also needs to be robust enough to go through harsh conditions of the field test.

1.3 Objectives of this research

The first objective of this study was to design the sensing system: Blueberry Impact Recording Device (BIRD). Firstly, the necessary hardware platform should be designed: a miniature sensor that can collect impact data by using tri-axis accelerometers, the sensor should be a standalone unit during data acquisition and it should be able to be connected with the computer for communication. Secondly, the BIRD system should also have a complete data acquisition and processing software.

The second objective of this study was to evaluate the bruising of blueberries during the mechanical harvest process in the field. The first aim was to quantitatively describe the process of a typical mechanical harvest of blueberries. The second aim was to use the BIRD sensor to evaluate the performance of three types of mechanical harvesters. The third aim was to correlate the sensor data with bruising rate of blueberries.

1.4 Organization of this thesis

This study is comprised of two main phases. The first phase focused on the design of the Blueberry Impact Recording Device (BIRD) system, including both the hardware and the software. The second phase of the study presented an application of using the BIRD system in the field to evaluate the three mechanical harvesters and blueberry bruising.

Chapter 2 focused on the hardware design of the BIRD sensing system. Three essential components (the BIRD sensor, the interface box and the computer software) were introduced and described. The calibration results and overall performance of the BIRD sensor was also discussed. Chapter 3 further explained the software development for the BIRD system, corresponding to the hardware components. The software was tested to verify its performance. In Chapter 4, the application of the BIRD sensing system in the field was discussed. Three major types of blueberry commercial mechanical harvesters were evaluated and compared. The mechanical harvest process was quantitatively described using the BIRD sensor along with a close-up video. The data from the BIRD sensor was also correlated with bruising probability of blueberries. The established model can be used for blueberry bruising prediction. The last chapter summarized the results of this research and provided a list of future studies.

CHAPTER 2

THE BLUEBERRY IMPACT RECORDING DEVICE (BIRD) SENSING SYSTEM: HARDWARE DESIGN

2.1 Overview

Bruising caused by the impact damage can occur frequently during mechanical harvest process for highbush blueberries. The overall goal of this study is to develop a sensor prototype to quantitatively measure the impact forces occurred on individual berry fruit during the mechanical harvest process. This information can be used to reduce bruising of blueberries during mechanical harvest. The sensing system developed in this study has three essential components: the sensor, the interface box and the software program on computer. The round circuit board of sensor is less than one inch (19.4 mm). It includes three Micro-Electro-Mechanical System (MEMS) accelerometers, one microcontroller along with other electronic components that have low power consumption. The sensor board and rechargeable battery are casted into a one inch (25.4 mm) sphere mold, using silicone rubber. The interface box serves as the intermediate communication platform to connect the sensor and the computer. The PC-software can acquire data via the RS232 communication and analyze data. The sensor was calibrated using centrifuge. The accuracy of the sensor output is -0.53% and 0.33% of the output span (±500 g). The resolution of the output is 1.465 (g) per digital count, with precision error 0.63%. By simulating the berry fruits under actual field situations, the sensor prototype could be a useful tool for evaluating the blueberry's bruising damage during the mechanical harvest process.

2.2 Introduction

US Blueberry industry has been growing rapidly in the past thirty years (Strik 2006). Traditionally, handpick is the primary harvest method for the fresh market, which was proven to be expensive and labor intensive (Takeda, Krewer et al. 2008). It poses a particular challenge when the labor source is volatile. Therefore, mechanical harvesters have been used to increase the overall efficiency of blueberry harvest. A serious existing problem of mechanical harvest is the excessive bruising of the berry fruit. It is estimated that 78% of the mechanical harvested blueberries are bruised, making them unmarketable (G.K. Brown 1996). In order to reduce bruising damages caused by mechanical impacts, it is important to identify and evaluate those impacts quantitatively. This information can be used to improve the current harvester and reduce bruising.

Various efforts have been made to measure bruising damages for large fruit and vegetables by using instrumented spheres which is also called "Pseudo fruits". An instrumented sphere is essentially a miniature wireless data logger packed in a sphere shape to mimic real fruits or vegetables in a packing line. Based on the purpose of sensing and type of data, there are two types of instrumented sphere sensors: one to measure dynamic impacts using accelerometers and the other to measure static load using pressure sensors. One well known example of the first class is the IS100 (Klug, Tennes et al. 1987; Simami, Tennes et al. 1987; Zapp, Ehlert et al. 1990; Bajema 1995). It had a tri-axial accelerometer, which could provide information about vector accelerations and variation of velocities. The first prototypes of the IS100 were a cube of 120 mm or a sphere of 140 mm diameter. The second generation of this sensor was an 89-mm sphere with improved accuracy (Zapp, Ehlert et al. 1990). A latest version of IS100 is called Impact Recording Device (IRD) with a smallest size of 57 mm in diameter (Techmark Inc, Lansing, MI). A more recent design is an acceleration measuring unit with a smaller size (13×13×42 mm) but in a rectangular block shape (Herold, Truppel et al. 2005; Geyer, Praeger et al. 2009). The unit can be embedded into actual fruits like potatoes to measure dynamic impacts. However, the sensor is not a sphere shape and the size is too big to be embedded into any blueberry fruits. Another instrumented sphere sensor is called PTR100 (its advanced version PTR200) which was designed and applied in evaluation of potato damages (Canneyt, Tijskens et al. 2003). The PTR sensor has an elongated semi-ellipsoidal shape in which sensors are mounted in different positions. Compared with spherical shape of the IRD, the shape of the sensor may be considered as an improvement towards the dynamic behavior of the fruits under practical measuring conditions since it has close physical shape. However, the sensor has only 100 Hz in sampling frequency and the size of sensor is large (53×53×83 mm).

The second type of the instrumented sphere is to measure the static load with pressure measuring sensors. The Pressure Measuring Sphere (PMS) with wireless telemetry has been developed and widely applied (Herold, Truppel et al. 1996). It is a 62 mm rubber sphere containing electronic components inside and weights 180g. The effective output of the PMS is under the assumption that damage of fruits and vegetables can be considered as accumulated result of multiple mechanical loading, including both static and dynamic loads. The sensor has the advantage of recording static loads, but it is not suitable to measure impact damages of blueberries with a much smaller size. Table 2.1 specifies the technical details about the three major types of sensors.

To our knowledge, no electronic sphere sensor has been developed and applied for small fruits like blueberries. The main challenge lies in the small size of the berry fruit (<30 mm). There are other application constraints, for instance, the IRD can't be dropped to a hard surface with more

Table 2.1 Compare of three popular instrumented sensors

Sensor Name	Shape/Size	Weight (g)	Data format	Sampling speed	Power supply	Maximum drop height to hard surface
IRD	Sphere Φ57mm	89	Tri-axes acceleration ±500 (g),3.4g Resolution	5KHz	6V DC Ni-MH 120mAH	10 cm
PMS60	Sphere Φ62mm	180	Pressure 0~100 N Static 0~400 N Dynamic	10KHz	5 cells Ni-MH 60mA	100 cm
PTR200	Ellipsoid 53×53×83 mm	170	Tri-axes Acceleration ±150(g)	100Hz	3.6V Lithium 400mAH	N/A

than 10 cm height, while blueberries can drop from 180 cm height to a hard surface during the mechanical harvest process. Therefore, in order to measure impact damages in blueberries, it is necessary to design an instrumented sphere that has a much smaller size than current commercially available instrumented sphere sensors but with sufficient sensing range. The sensor also needs to be robust enough to go through harsh conditions of the field test, so the sensor can be specifically used for blueberry mechanical harvest process.

Two objectives of this development were to:

1. Design a miniature instrumented sphere sensor that can be specifically used for blueberry mechanical harvest evaluation.
2. Calibrate and characterize the sensor to validate its performance.

2.3 Structure of the BIRD

As shown in Figure 2.1, the Blueberry Impact Recording Device (BIRD) consists of three components: the BIRD sensor, the interface box, and the software program on the PC for data retrieving and analysis (i-BIRD). The BIRD sensor is primarily used for impact data collection and recording. Its electronic circuit board, components, and a rechargeable battery were cast in a one-inch sphere by silicon rubber. The sensor is connected to the interface box with a five pin connector, to set up the I^2C communication and power supply. The interface box communicates with the PC via RS232 cable. Users can configure and download data from the BIRD sensor by using the i-BIRD software that was developed using the graphic programming language LabVIEW 8.2 (National Instruments, Austin,TX). Detailed descriptions of these components were provided in the following sections.

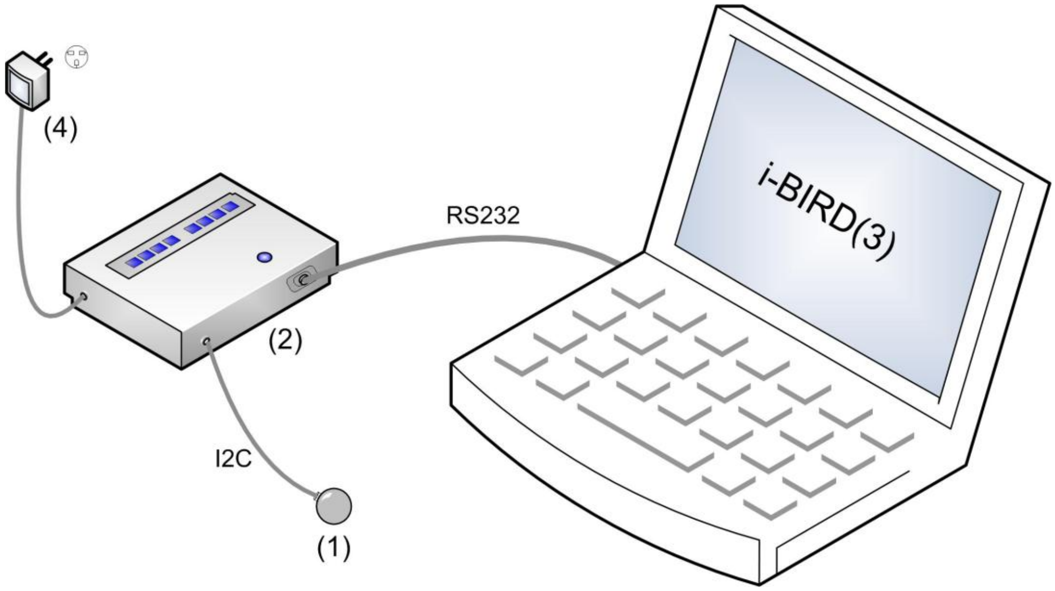


Figure 2.1. BIRD system diagram; BIRD Sensor (2) BIRD Interface box (3) i-BIRD Software (4) DC Power supply for the interface box

2.3.1 The BIRD sensor

Ideally, the size of the sensor would be comparable to the size of a typical blueberry (15-25 mm). Despite recent advances in electronic products, it turned out that it is extremely challenging to put all components onto a circuit board with 15 mm diameter. In addition, the availability of small size accelerometers with appropriate sensing range is another constraint in our design. Eventually, a 25.4 mm (one-inch) instrumented sphere was developed. Although the physical dimension is important, the electronic sensor circuit board should firstly maintain required data collection performance. Specifically, the sensor should reach the required data sampling frequency and accuracy. Preliminary test showed that the impact duration of blueberry lasts for 3~4 ms. It is measured by mounting an accelerometer on top of the blueberry fruit. Based on Nyquist Sampling theorem, a minimum 668 Hz sampling frequency is needed to avoid aliasing effect. The sensing range of the accelerometer should also be adequate to record all impacts. Preliminary laboratory test and previous literature indicated that ±500 g range is appropriate to record impact data from blueberries. Adequate power supply, accurate real time clock (RTC), and sufficient data storage space are also critical aspects of the BIRD sensor design. Specifically, the power should last at least for one hour in order to finish data collection in one row. Recharging is essential because the sensor housing can’t be opened and battery cannot be replaced. A memory size of 128 KB is desirable in order to record at least 347 impacts. Figure 2.2 shows the physical size and layout of the sensor. The round circuit board inside the sphere has a diameter of 19.4 mm (~0.75 inch). The sphere sensor after casting is 25.4 mm in diameter, as shown in Figure 2.3. With such a small size, the BIRD sensor can easily go through the blueberry mechanical harvest process.

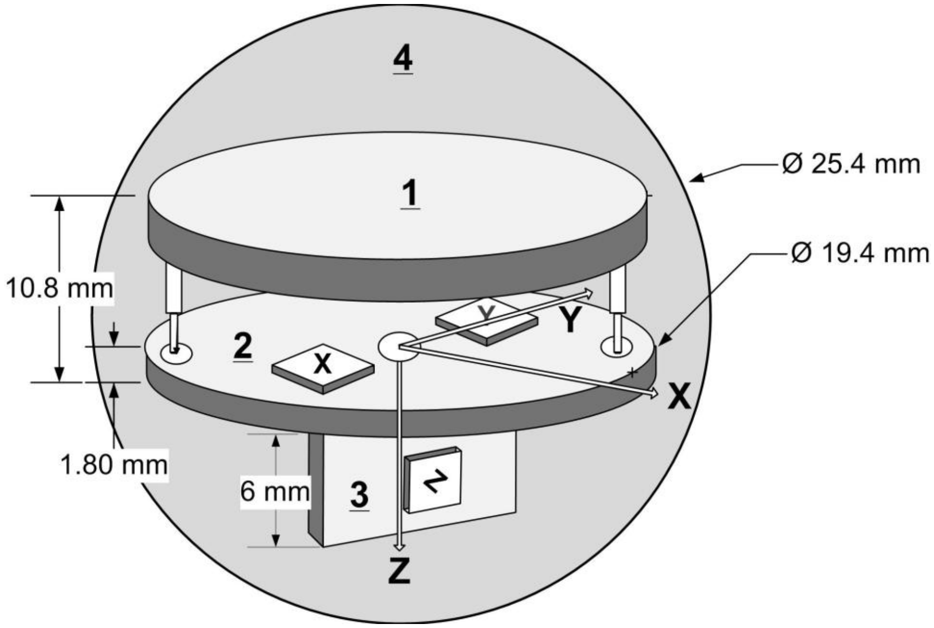


Figure 2.2. Inside structure of the BIRD sensor; 1.Rechargeable coin cell. 2. Main PCB board 3.Vertically mounted PCB board for Z axis accelerometer 4. Housing

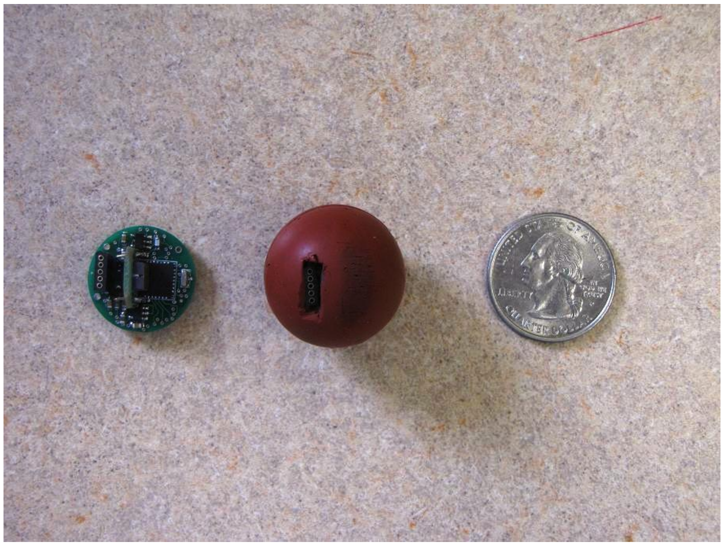


Figure 2.3. BIRD sensor circuit board and the cast sensor

The main function of the BIRD sensor is to collect impact data and store them into onboard memory. As shown in Figure 2.4, there are four essential components on the circuit board: a microcontroller, three single axis Integrated Microelectromechanical Systems (iMEMS) accelerometers, a 128 KB Ferroelectric Random Access Memory (F-RAM) chip for data storage, and a rechargeable coin battery for power supply. The selection criteria of these components and their performance are discussed below. The PCB circuit board was designed using PCB123 (V3.1, Sunstone Circuits, Mulino, Oregan). The primary design criteria of the PCB design is minimize board size. Layout of components on the top and bottom surfaces also requires careful organization. The layers of the PCB circuit board was increased from 2 to the finally 6, which enables the successful design of the final miniature round board, with diameter of 19.5 mm.

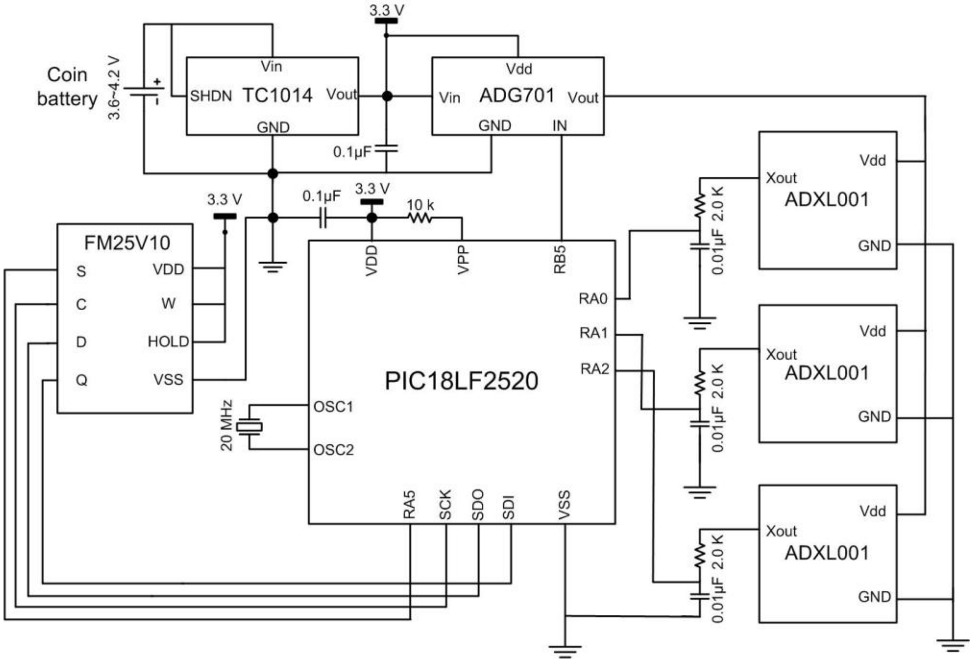


Figure 2.4. BIRD sensor schematic; (ADXL001: Accelerometer, ADG701: Analog switch, FM25V10: RAM memory, PIC18LF2520: MCU, TC1014: Voltage regulator)

2.3.1.1 Tri-axial accelerometer

Size and sensing range are primary concerns in selecting a right accelerometer. Other parameters, such as accuracy and precision, power consumption, and cost, are also important factors. Appendix D provides a list of popular MEMS accelerometers. Single chip tri-axes accelerometers with ±500 g sensing range, small size and low power consumption were not readily available. Therefore, three single-axis accelerometers were combined to make a tri-axial accelerometer. A fifth-generation iMEMS chip ADXL001 (Analog Devices, Norwood, Massachusetts) was selected for this design. This accelerometer provides ±500 g dynamic range with resolution of 2.2 mV/G, maximum 22 KHz resonant frequency, high linearity, 5000 g impact survival range, and a 5 × 5 × 2 mm package size. This sensor can be operated at low voltage (3.3 V) and has low power consumption (2.5 mA), which is critical in this design. The X and Y sensing accelerometers were mounted on the surface of the main PCB board with their sensing axes perpendicular to each other. The third accelerometer was mounted vertically to the surface of the main PCB board as the Z axis. Later calibration tests validated that this method of mounting can provide accurate output.

2.3.1.2 Microcontroller unit (MCU)

The MCU's operation speed, physical dimension, power consumption and its I/O resources are important factors that need to be considered. As the Central Processing Unit of the BIRD sensor, the MCU collects analog acceleration signals from the accelerometer, converts them into digital values and serially writes them into the memory. It also manages power and communicates with the interface box. The PIC18LF2520 (Microchip, Chandler, AZ) microcontroller was selected and tested based on the above mentioned criteria. The chip has 28 pins with 25 I/O pins and 5 channels of 10-bit AD conversion. With 20 MHz oscillator option, the chip can execute up to 5 million instructions per second (MIPS). Working voltage ranges from 2.0~5.5 V and the current drain of the MCU in sleep mode can be down to 0.1 μA at 25 ℃ thanks to the nanoWatt technology. The chip has 32 KB program memory, and 1536 bytes of Static Random Access Memory (SRAM). The large RAM memory enables the buffer to store more array values for recording impact data. It also supports the I^2C (Inter-Integrated Circuit), the 3-wire SPI (Serial Peripheral Interface), and USART (Universal Asynchronous Receiver/Transmitter) communication protocols. The size of the chip is 6 × 6 × 1 mm with a Quad Flat No Leads (QFN) package.

2.3.1.3 Memory

Primary criteria for memory chip selection are operation speed, communication protocol, package size, and power consumption .The F-RAM (FM25V10, Ferroelectric Random Access Memory) (RAMTRON, Colorado Springs, CO) serial memory with 128 KB capacity was selected. It has high writing speed (SPI, 40 MHz), nonvolatile function and low power consumption (0.3~3 mA). The physical dimension is 4.9 × 6.0 mm. Besides F-RAM, two other types of memories were also compared: the Electrically Erasable Programmable Read-Only Memory (EEPROM) and the Static Random Access Memory (SRAM). Available EEPROM chips can store up to 128 KB data, and also have competitive power and size options. One weakness of the EEPROM chips is its slow writing speed caused by the hardware delay, which takes an additional 5-10 ms for each writing cycle other than the bus writing period. The SRAM traditionally operates at high bus speed, but it does not provide non-volatile feature, which may cause data loss due to potential battery failure.

2.3.1.4 Power supply

Selecting the right battery is one of the most challenging tasks in this design due to size constraint. The battery should provide a stable power source with physical dimensions compatible with the electronic circuit board. It should also have abundant capacity to keep the sensor working fully for three hours to ensure valid data collection. A 3.7 V, 75 mAH Lithium Ion rechargeable coin battery (PD2032, Korea PowerCell Inc, Chungnam, Korea), meets those requirements. A linear Low Voltage Drop (LDO) DC-DC converter (TC1014, Microchip, Chandler, AZ) was applied to regulate the battery voltage to 3.3V. The sensor's power modes include sleep mode, standby mode, and active mode to accommodate different working status. In sleep mode, power supply to three accelerometers is cut off, and the MCU is also in sleep mode. This mode can be used during recharging period or when the sensor is waiting to be used. In this mode, current drain of the whole sensor is limited to 0.35 mA. The standby mode enables the MCU to establish its communication with the interface box, while the three accelerometers are power off, which has a current drain of 5.6 mA. During active mode, all components are powered and the sensor is in active data recording period. Under this mode, the maximum current drain is12.95 mA. Switching between the power modes is achieved by selectively activating the microcontroller's I/O pins and peripheral circuitry. For instance, power supply to the three accelerometers is controlled by an analog switch (ADG701, Analog Devices, Norwood, Massachusetts) as shown in Figure 2.4, which can be toggled by an I/O pin (RB.5 in Figure 2.4) of the MCU. The battery can be recharged by the external recharger in the interface box during field tests. The recharger was built upon a recharging chip MCP73831 (Microchip, Chandler, Arizona) which is powered by a 9.0 V battery or a 9.0 V DC input.

The discharging and recharging performance of the BIRD sensor is provided in Figure 2.5 (a) and (b), respectively. The voltage is about 4.1 V when the battery is fully recharged. During the active data sampling process, the voltage of the battery drops below 3.3 V after approximately 4 hours of stable operation. The battery voltage is regulated to 3.3 V to provide a stable power source for the sensor's components. To prevent data loss, a low voltage detection function can be realized by sampling the battery voltage periodically through one of I/O pins of the microcontroller in the interface box. A notice of low battery will be provided via the interface board when the battery cut off voltage is reached.

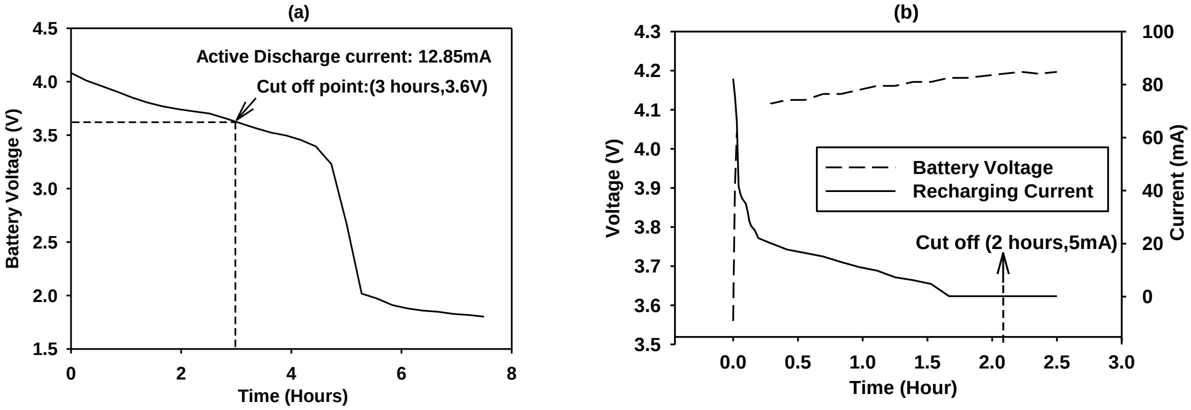


Figure 2.5. Battery performance of the BIRD sensor battery; (a) Discharge (b) recharge

2.3.2 Cost of the sensor

Cost of the BIRD sensor and interface box was estimated in the Table 2.1. The prices were the regular retail value. Recursive engineering cost can be much less. Comparing with commercial price, for example, the IRD can cost up to 5000 $, while the BIRD cut the cost to 8% of it.

Table 2.2. Cost of the BIRD sensor and the interface box

Component	Model Number	Price($)
MCU	PIC18LF2520	7
Accelerometer	ADXL001(3)	150
F-RAM	FM25V10	12
Analog switch	ADG701	1.68
Battery	PD2032	12
Regulator	TC1014	0.41
PCB	PCB Board	100
MCU	PIC16LF877A	7
RS232	MAX232	3.3
Parallel LCD	LCM-S02002DSR	15.82
F-RAM	FM24V10	12
Recharging chip	MCP73831	0.68
Resistors, Capacitors	-	10
Others	Connectors	25
	Subtotal	356.89

2.3.3 Sensor housing design and features

Three types of housing materials were selected and tested: epoxy resin, silicone rubber (Shore A60) and silicone rubber (Shore A30). Preliminary tests showed that harder molding materials have better performance in keeping data consistent when the sensor experiences impacts from different angles. Epoxy Resin is one kind of hard material that has been selected by other impact recording devices. However, our preliminary tests showed that the sensor was exposed to excessive high impact values when epoxy resin was used due to its hard surface. This could limit the drop height of the sensor to a relatively low level to avoid the damage of the sensor, while the drop height can go up to two meters during blueberry mechanical harvest process. For example, the IRD can't be dropped to a hard surface over 10 cm. Silicone A30 has properties similar to a rubber band, while Silicone A60 is similar to a car tire tread. The silicone rubber with Shore A60 was selected due to its adequate hardness to keep data consistent, to prevent circuitry damage, and to achieve the required drop height (~2 m) during the harvest.

2.3.4 Interface box

The interface box was designed as an intermediate device to connect the BIRD sensor with the computer. As illustrated in the schematic in Figure 2.6, the interface box includes a microcontroller (PIC16F877, Microchip, Chandler, AZ)), memory, an LCD, RS232 communication port and the recharging circuits to recharge the battery on the BIRD sensor. A 12 V power supply to the interface box is provided by a DC adapter. A parallel liquid crystal display (LCD) is used to display operation status of the BIRD sensor. An external FRAM (I2C) with 1 Mb memory capacity is used to store data downloaded from the BIRD sensor. The interface box also has switches to control the board power supply to the recharging circuits. The recharging circuit uses the MCP73831 chip from Microchip, which provides a recharging voltage of 4.2 V, 100 mA or 500 mA recharging current based on user selection. Communication between BIRD sensor and the interface box employs two-wire asynchronous serial communication (I2C); communication between the interface box and the computer uses RS232 protocol.

2.3.5 Computer Software (i-BIRD)

i-BIRD computer software has three major functions: the BRID sensor configuration, data downloading, data processing and management. The configuration of BIRD sensor includes user selectable sampling frequency and threshold acceleration, synchronization with computer time, and control of power modes (sleep and active). Data can be downloaded from the sensor to the interface box and then uploaded to PC. Data on the FRAM memory chips of both the sensor and the interface box can be erased by sending corresponding commands from the computer. After data downloading, the i-BIRD can convert the raw data from digital counts to real acceleration

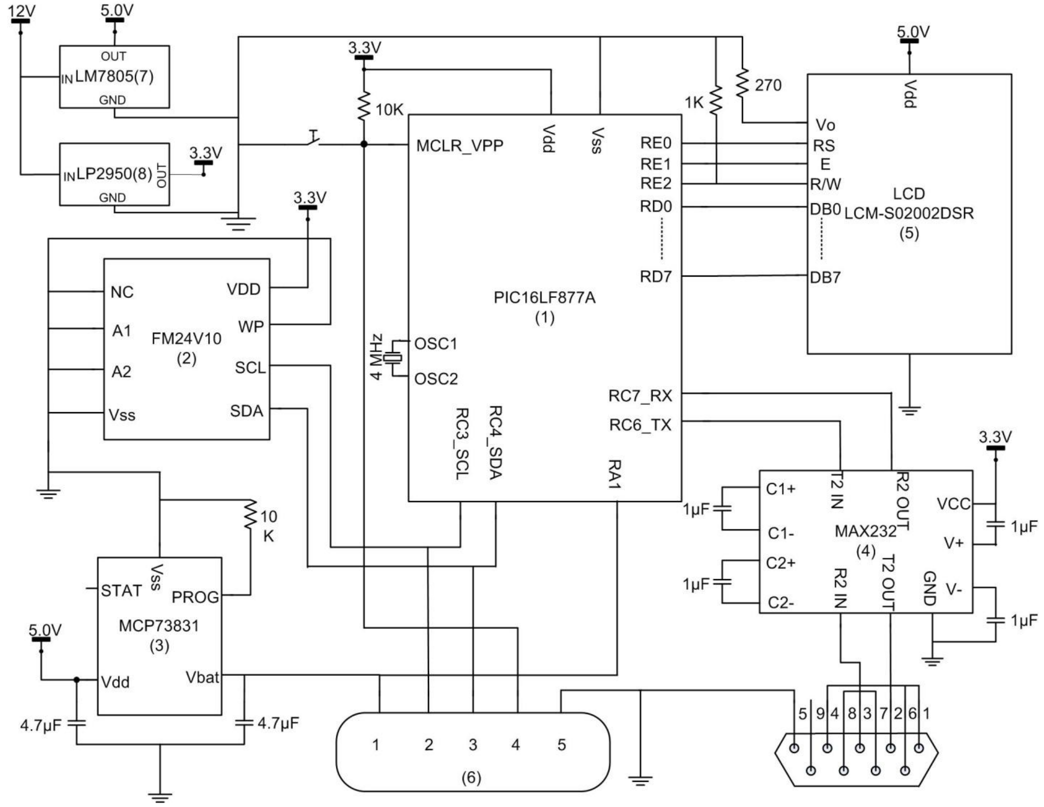


Figure 2.6. Schematic of interface box circuit; (1) The MCU (2) Memory chip (3) The recharging chip (4) The RS232 communication chip (5) LCD Monitor (6) Five pin connector:1> Recharging power source 2> I2C clock Pin 3> I2C Data pin 4> Reset pin for sensor 5> GND (7) 5.0V DC voltage regulator (8) 3.3V voltage regulator

values, save the processed data into spreadsheet files, and display the impact data graphically. The detailed description of the i-BIRD software program will be introduced in a later chapter.

2.3.6 Data flow and management

Figure 2.7 shows the structure of data flow between components in the BIRD system. The microcontroller in the BIRD sensor samples three channels of analog acceleration signals, and converts them into digital values via the analog to digital conversion. Acceleration values that are greater than the user-defined threshold are recorded on the onboard F-RAM via the SPI communication protocol, in order to achieve high sampling frequencies. The sensor uploads recorded data to the interface board by writing data into the F-RAM chip on the interface board, via the I2C bus. The MCU on the interface board reads the data from the F-RAM via the I2C protocol and then sends them to the computer through the RS232 cable.

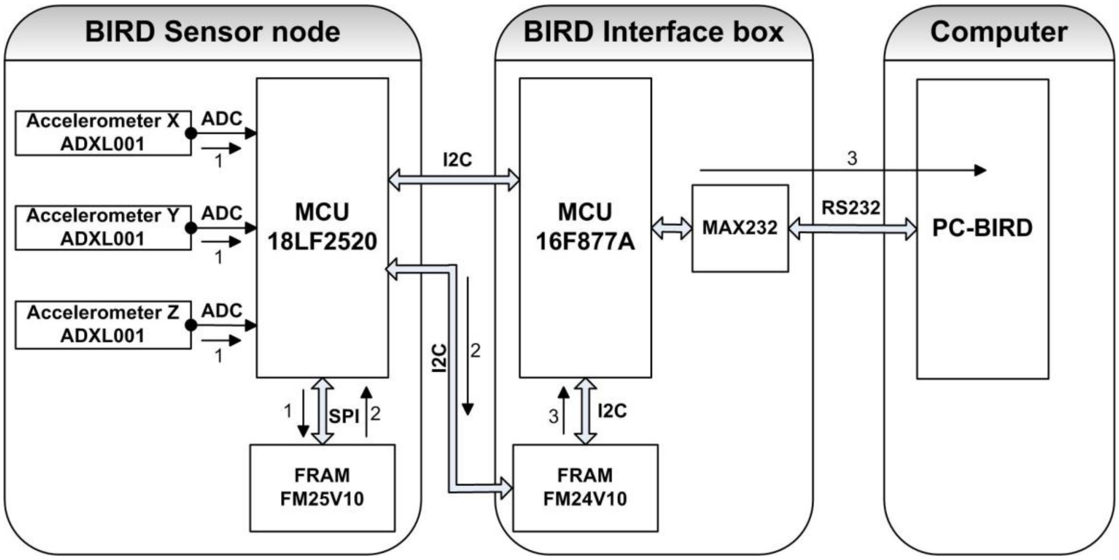


Figure 2.7. BIRD data flow and communication; 1.Sample accelerometer data and record data into sensor memory 2. Read data from sensor memory and write data into the interface memory. 3. Read interface box memory and upload data computer

2.4 BIRD Calibration

2.4.1 Calibration using centrifuge

The BIRD sensor was calibrated using a centrifuge (EPPENDORF 5810, Hamburg, Germany) following ANSI/ISA,1979(Wheeler and Ganji 1996) . The principle of using a centrifuge for calibration is that it can provide accurate and stable acceleration at the centripetal direction at a given rotation speed. As illustrated in Figure 2.8, if the distance between a certain point (the sensor in this case) to the pivot of the centrifuge is known, the acceleration at that point can be calculated by the following equation:

$$G = (\frac{RPM \times 2\pi}{60})^2 \times r \qquad (m/s^2) \qquad (2)$$

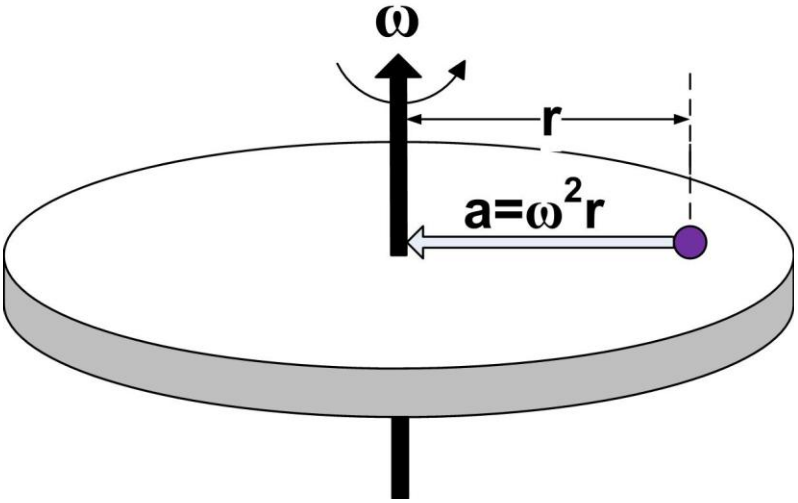


Figure 2.8. Principle of BIRD sensor calibration using a centrifuge; (ω=angular speed; r=distance from the center of the sensor to the pivot; a=centripetal acceleration)

The mounting orientation of the BIRD sensor was not purposely controlled in each test because the summation of acceleration values from three axes instead of from single axis was calibrated.

Eleven rotational speeds of the centrifuge were selected from 440 to 1960 rotations per minute (RPM), corresponding to centripetal acceleration values from 25.28 to 501.67 g (g= 9.80 m/s^2). This range is adequate to calibrate our accelerometers which have the dynamic range of ±500 g. The centrifuge was started at lowest speed level, stepped up to the highest speed, and then gradually slowed down to the lowest speed. This up and down cycle was repeated six times. In order to avoid the error that could be potentially created at the starting and ending period of the centrifuge, the first five acceleration values of the first cycle and the last sixteen accelerations of the sixth cycle were not used, to make five complete cycles. Due to the up and down cycles, there were ten replicates for each of the 11 acceleration values except for the maximum acceleration value 501.67 g which had only 5 replicates. Those data were used for a linear regression analysis between the acceleration values created by the centrifuge and recorded by the BIRD sensor. As illustrated in Figure 2.9, an extreme linear relationship between the centrifuge and BIRD sensor was observed with a coefficient of correlation (r^2) of one and standard error of prediction of 0.5958 g. Given the sensing range of 500 g from each accelerometer, this error was minimal.

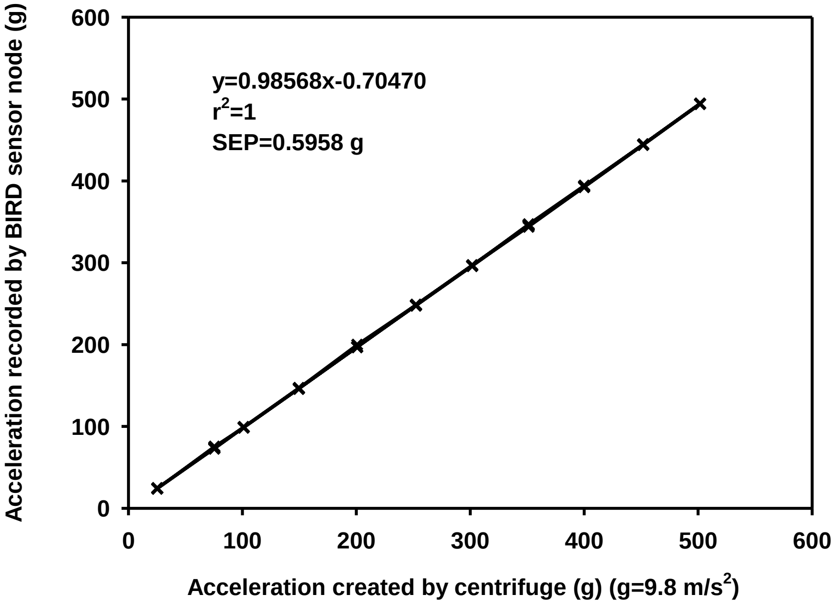


Figure 2.9. Linear regression of acceleration values generated by the centrifuge and the BIRD sensor

To further evaluate the accuracy of the BIRD sensor data, the difference between the measured values by the BIRD sensor and the values calculated by the best-fit equation was analyzed. As shown in Figure 2.10. The accuracy limit, which is the range of deviation from both the up and down cycles, is from -2.60 g to 1.62 g in this test. Given an input range of 0~500 (g), the output span of the prediction line is from -0.7 g to 492.2 g, produces a span of 492.9g. The accuracy becomes -0.53% and 0.33% of the output span.

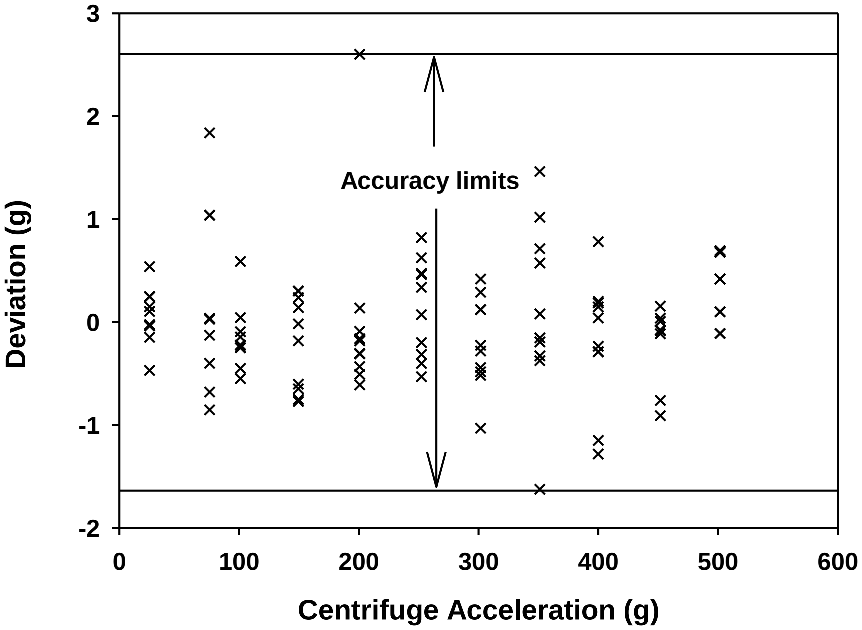


Figure 2.10. Deviation of acceleration values measured by the BIRD sensor from the centrifuge acceleration values.

The repeatability error, which is also the estimated precision uncertainty, can be evaluated from the maximum variability of successive measurements of the same value input. In Table 3, the last column lists the maximum variability of each measurement. The maximum value is 3.11 g, corresponding to a precision uncertainty of 0.63% in the prediction range.

The hysteresis error is evaluated as the maximum difference between the “up” cycle and the “down” cycle for the same input value in any of the calibration cycles. In Table 2.3, the maximum difference is found to be 3.21 g and occurs in the replicate 5 at 200.79 g, which corresponds to 0.65% of the output span of the predicated line.

Table 2.3. Deviation of acceleration values measured by the BIRD sensor from acceleration values created by the centrifuge

Centrifuge Acceleration (g)	BIRD Rep#1 (g)	BIRD Rep#2	BIRD Rep#3	BIRD Rep#4	BIRD Rep#5	BIRD Rep#6	Average of cycles (g)	Maximum variability (g)
25.2819		-0.0250	0.1484	-0.4700	-0.0370	-0.1470	-0.1063	0.6181
75.4279		1.8372	1.0375	-0.4010	1.0392	-0.1280	0.6771	2.2382
101.1278		-0.2270	-0.1450	-0.2510	-0.2220	0.5886	-0.0513	0.8400
149.5108		0.3005	0.1408	-0.0170	0.2379	0.3029	0.1929	0.3204
200.7929		-0.1620	-0.090	0.1364	2.6014	-0.5080	0.3955	3.1094
252.3101	0.8200	0.4731	0.0709	-0.5310	0.4614		0.2588	1.3514
301.7117	-0.2820	-0.5170	-0.2260	-0.4430	0.4179		-0.2100	0.9347
351.2309	1.4621	1.0177	-1.6250	-0.1530	-0.1930		0.1017	3.0868
399.9274	0.7808	0.2013	-1.1500	-0.2360	0.1473		-0.0513	1.9311
451.7841	-0.0780	-0.083	-0.7610	0.1542	0.0375		-0.1460	0.9155
501.6689	0.4187	0.6936	0.1000	-0.1120	0.6781		0.3558	0.8052
451.7841	-0.0830	-0.0830	-0.9100	0.0066	-0.1130		-0.2365	0.9171
399.9274	0.0400	0.1850	-1.2820	-0.2900	-0.2870		-0.3267	1.4668
351.2309	0.5735	0.7136	-0.3750	0.0800	-0.3280		0.1328	1.0887
301.7117	0.2894	0.1183	-1.0310	-0.4840	0.1200		-0.1973	1.3200
252.3101	0.3383	0.6241	-0.3140	-0.2000	-0.4020		0.0091	1.0266
200.7929	-0.3040	-0.3100	-0.4320	-0.1820	-0.6110		-0.3680	0.4283
149.5108	-0.6030	-0.752.	-0.6490	-0.7710	-0.1830		-0.5917	0.5880
101.1278	0.0433	-0.0940	-0.4500	-0.5500	-0.2500		-0.2602	0.5930
75.4279	0.0344	0.0371	-0.6800	-0.8540	0.0278		-0.2868	0.8908
25.2819	0.5377	0.2449	0.2481	-0.0380	0.1033		0.2192	0.5758

2.4.2 Dynamic Dropping test

It is important to verify the performance of the BIRD sensor under a situation with dynamic impacts. For this purpose, the BIRD sensor was tested using an impact table, as shown in Figure 2.11. When the impact board was released at a given height (distance between the impact point and the base) and collided with the hard wood base, an impact was created and recorded by the sensor mounted on the impact board. The impact board was released at five different drop heights (15CM,30CM,45CM,60CM,75CM), with 20 replicates for each drop height. Each drop created multiple impact values (including rebounds), but only the maximum impact values were evaluated for each drop.

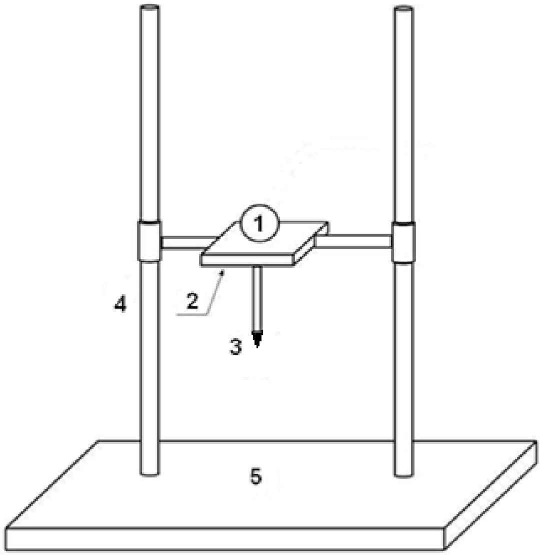


Figure 2.11. Mounting the BIRD sensor on the impact table; 1 BIRD sensor 2 Impact Board 3 Impact material(Sharp wood tip) 4 Sliding Tracks 5 Base

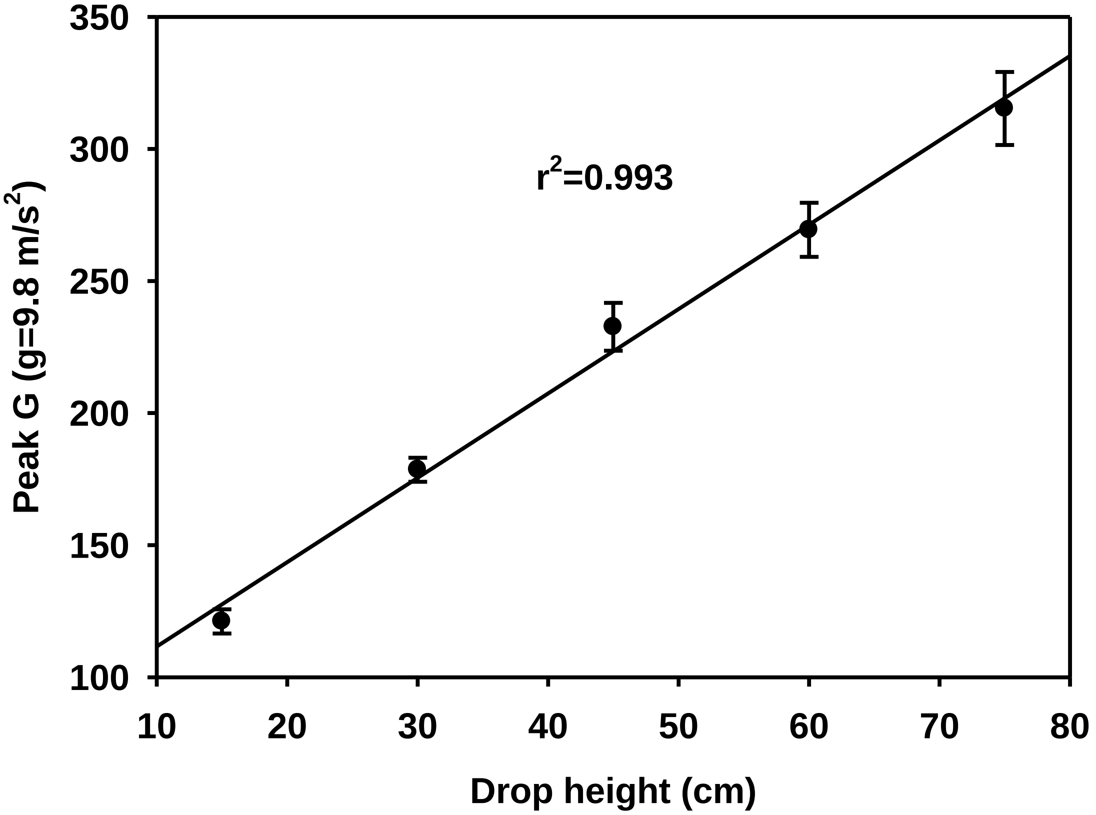


Figure 2.12. Dynamic drops peak G recorded by BIRD sensor

As demonstrated in Figure 2.12, the Peak G recorded can increase along with drop height with outstanding linear relationship (r^2>0.98). The standard deviations are 4.57, 4.54, 9.07, 10.23, and 13.83 for the five drop heights from 15 cm to 75 cm respectively. Variation can also come from the impact table's release heights, the consistency from impact material's contact with the base. Considering all those error sources, the BIRD sensor still maintains good consistency.

2.5.Conclusion

In this research, a miniature impact recording sensor was developed to evaluate the blueberry mechanical harvest process. The sensor can collect dynamic impacts with acceleration range of ±500 g and 3.0 KHz sampling frequency. The sensor is sealed into a one inch silicone rubber sphere. Calibration results showed an accuracy of -0.53% and 0.33% over the range of ±500 g, with precision error of 0.63%. It also has rechargeable battery that can work fully for three hours. An interface box was designed to provide interconnection between BIRD sensor and the computer, so data can be downloaded into computer for further analysis. The initial design requirements are met based on the current performance. With all those features, The BIRD system can be a useful tool to evaluate the mechanical harvest process of blueberry. Future work will be to improve the sensor's robustness for field application.

CHAPTER 3

THE BLUEBERRY IMPACT RECORDING DEVICE (BIRD) SENSING SYSTEM: SOFTWARE DESIGN

3.1 Overview

This chapter presents a complete impact data acquisition, processing and analyzing software system that is used by the hardware of the Blueberry Impact Recording Device (BIRD). The software system has three major sections corresponding to hardware: The BIRD sensor program, the interface box program and the computer software (i-BIRD). The software on the microcontroller of the BIRD sensor manages data sampling and recording from three accelerometers, with maximum sampling rate of 3.0 KHz. The program on the interface box was developed to establish the communication between the BIRD sensor and the computer. Users can configure the sensor via the i-BIRD computer software developed in LabVIEW environment, with different options of sampling frequency (682~3050 Hz) and thresholds (0~205 g). The acceleration data on the BIRD sensor can be downloaded, processed and graphically displayed on the computer. The performance of the software was tested in terms of sampling rate, impact shape distortion and timer accuracy. The accuracy of the real time clock on the BIRD sensor was calibrated with error of 0.073%, which provides an accurate timer for the recorded data. The impact curve shape recorded by the BIRD matched well with the impact curve recorded by a reference high frequency (10 KHz) data logger. The relative error of the velocity change calculated by the program on the BIRD sensor was less than 5% compared to the reference data from the data logger. The software system developed in this chapter enabled the BIRD sensor function properly and to be a useful tool to collect impact data during the blueberry mechanical harvest.

3.2 Introduction

Successful studies have been reported in literature about using “artificial fruits” to identify the source of produce damage during mechanical harvest, post harvest and handling process. The “artificial fruits” are subjected to the same load experiences as the real fruits. The recorded signal responses of the instruments can be used to identify critical points that exceed the damage boundary of fresh produce. Available commercial instrumented sphere sensors can be categorized into two major types: sensors that collect acceleration data or sensors that collect pressure values. The Impact Recording Device (IRD) is the popular sensor that records acceleration data and has been used in many studies (Techmark Inc, Lansing, MI). As the latest version of the previous IS100 (Klug, Tennes et al. 1987; Simami, Tennes et al. 1987; Zapp, Ehlert et al. 1990; Bajema 1995), the spherical IRD has a diameter of 57 mm and records three axes of acceleration values. The acceleration data are saved as single impacts. The acquired impact data provides information about vector summation of accelerations and variations of velocities. The IRD can set a sampling threshold that filters impacts with peak G less than the specified value. The maximum sampling frequency reached 4.2 KHz. The IRD kit also has computer software (PCIRD) for configuration of the sensor, downloading and processing data, and generating reports. Another popular instrumented sphere sensor is the PTR 100, or PTR200, developed by researchers in Denmark (Canneyt, Tijskens et al. 2003). The PTR sensors improved the instrumented sphere design by simulating the shape of a real potato (53×53×83 mm). It has an ellipsoidal body and can sample acceleration data at 100 Hz, which was too low to record the peak G of high frequency impacts. Another type of instrumented sphere sensor, such as the PMS-60, was used to record both the static pressure loads and dynamic impacts. The PMS-60 weighs 180g and the pressure sensors were cast in a rubber sphere (Rubber, Shore A80) with 62 mm in diameter (Herold, Truppel et al. 1996). The sampling software of the PMS-60 reaches 10 KHz in sampling frequency, which can register rapid loading of forces, including both static compression and dynamic impacts. Recorded data also has the "leader" and the "trailer" to provide more information for each impact and compression. A real time clock that can be synchronized with PC was created to provide time for the data. A program that runs under the MS-DOS was designed to control the sensor sphere, display and process the data.

A miniature sensor that is comparable to the blueberry in size was developed in this research. The sensor (Blueberry Impact Recording Device, BIRD) was intended to record impacts during the blueberry mechanical harvest process. The BIRD system has three separate parts: a one-inch sphere sensor for data sampling and recording, an interface box which connects the sensor with the computer, and the computer for sensor configuration, data downloading and processing. This chapter describes the details of the software system development for the BIRD system. Corresponding to the BIRD hardware, the software also has three components: the BIRD sensor program, the interface program, and the computer software i-BIRD.

The objectives of this software system were to:

1. Design programs for the microcontrollers at the BIRD sensor and the interface box.
2. Develop the i-BIRD computer software to configure the sensor, download data, process and display data graphically.

3. Test the software for its sampling rate, impact shape distortion, timer accuracy, and the overall speed and efficiency.

3.3 Description of the BIRD hardware system

The BIRD hardware platform has three parts: the BIRD sensor, the interface box, and the computer. The BIRD sensor is cast in a silicone rubber as an “instrumented sphere” with 25.4 mm (1 inch) in diameter and 14 grams in weight. Inside the sphere is a round circuit board. The circuit board consists of a microcontroller (PIC18LF2520, Microchip, Chandler, Arizona), three accelerometers placed in orthogonal directions (ADXL001, Analog Devices, Norwood, Massachusetts), one Ferroelectric-Random Access Memory (F-RAM) (FM25V10, RAMTRON, Colorado Springs, CO), a Lithium Ion rechargeable battery, and other conditional circuits. The microcontroller’s three 10-bits analog-to-digital(A/D) converter samples the three axes of accelerometers at a maximum frequency of 3.0 KHz, and stores the data into the 128 KB F-RAM. The BIRD sensor connects the interface box with a five pin connector, with two of them for the I^2C communication, and three others for recharging power source, reset of sensor MCU and ground. The interface box provides the interface between the BIRD sensor and computer through RS232 and I2C communication, as well as recharges battery in the BIRD sensor. The interface circuit board has one MCU (PIC16LF877A), an RS232 communication port and the recharging circuits. The LCD on the interface box displays the working status of the BIRD sensor when it is connected with the interface box.

3.4 BIRD software overall description

Software of the BIRD system consists of three parts that correspond to the three components of the BIRD hardware: the sensor program, the interface box program, and the computer graphical user interface program (i-BIRD).The PICBasic Pro (PBP, MicroEngineering Labs, Colorado

Springs, Colorado) compiler was used for program development of microcontrollers on the BIRD sensor and the interface box. Sensor program codes were programmed into the microcontrollers via the in circuit serial programmer (ICSP programmer, MicroEngineering labs, Colorado). The graphic user interface program on the computer was designed in the LabVIEW 8.2 environment (National Instruments, Austin, Texas).

The overall software structure of the BIRD system is shown in Figure 3.1. The BIRD sensor program samples data from three axes of accelerometers and records them in its onboard RAM. It communicates with the microcontroller on the interface box via the I2C communication, using two wires. The interface box program transfers commands from the computer to the sensor, and uploads data from the sensor to the computer via a RS232 cable. The i-BIRD computer software provides a graphical user interface for sensor configuration, data download, and data processing. Data can also be graphically displayed for quick view of the test result.

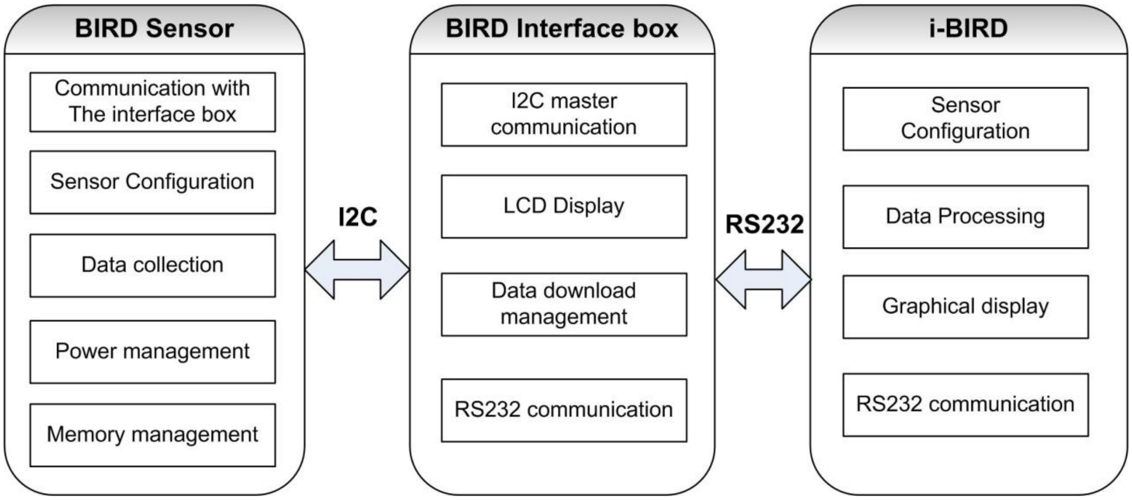


Figure 3.1. Overall software structure of the BIRD system

3.4.1 The BIRD sensor program

The BIRD sensor program has three main functions: sensor configuration, impact data sampling, and data management. The program diagram is shown in Figure 3.2. The sensor configuration is processed by receiving commands from the interface MCU, when the sensor MCU works in I2C slave mode. By executing each command, the sensor program enters into the corresponding subroutine, which carries the specific function. The whole program occupies 5.36 KB of the total 32 KB program memory in the microcontroller. Figure 3.3 shows the hardware resources that are relevant to the BIRD sensor program: three of 10 bits A/D channels samples the X, Y and Z axis respectively. With simple configuration, the Master Synchronous Serial Port (MSSP) was configured firstly into I2C mode to communicate with the interface MCU and the interface box memory (I2C-FRAM). The I2C communication requires only two wires but can connect with multiple components with a given address. The MSSP module is also configured to hardware SPI module to communicate with the sensor SPI-FRAM memory. The USART (Universal Synchronous/Asynchronous Receiver/Transmitter) is setup to establish the RS232 communication protocol. The MCU works at 5 MIPS based on 20MHz resonator frequency. The Timer 2 module was employed to create the Interrupt Service Routine (ISR) for the sensor's Real Time Clock (RTC).

3.4.1.1 Sensor configuration

The sensor can be configured with users' selections. The BIRD sensor is configured by receiving commands that are sent out from the computer and transmitted by the interface MCU. The communication between the two MCUs of the BIRD sensor and the interface box employs the I2C protocol, as shown in Figure 3.4. The interface box MCU (Master) sends control commands to the MCU of the BIRD sensor in pre-defined format. Each command comprises 10 digits, with the first 9 digits contain the command information and the last digit determines the command type. The sensor MCU executes each command received from the interface box, which can set up different power modes (sleep, standby, and active modes), sensor sampling threshold, sampling frequency, and sensor clock synchronization with the computer clock. Setup of

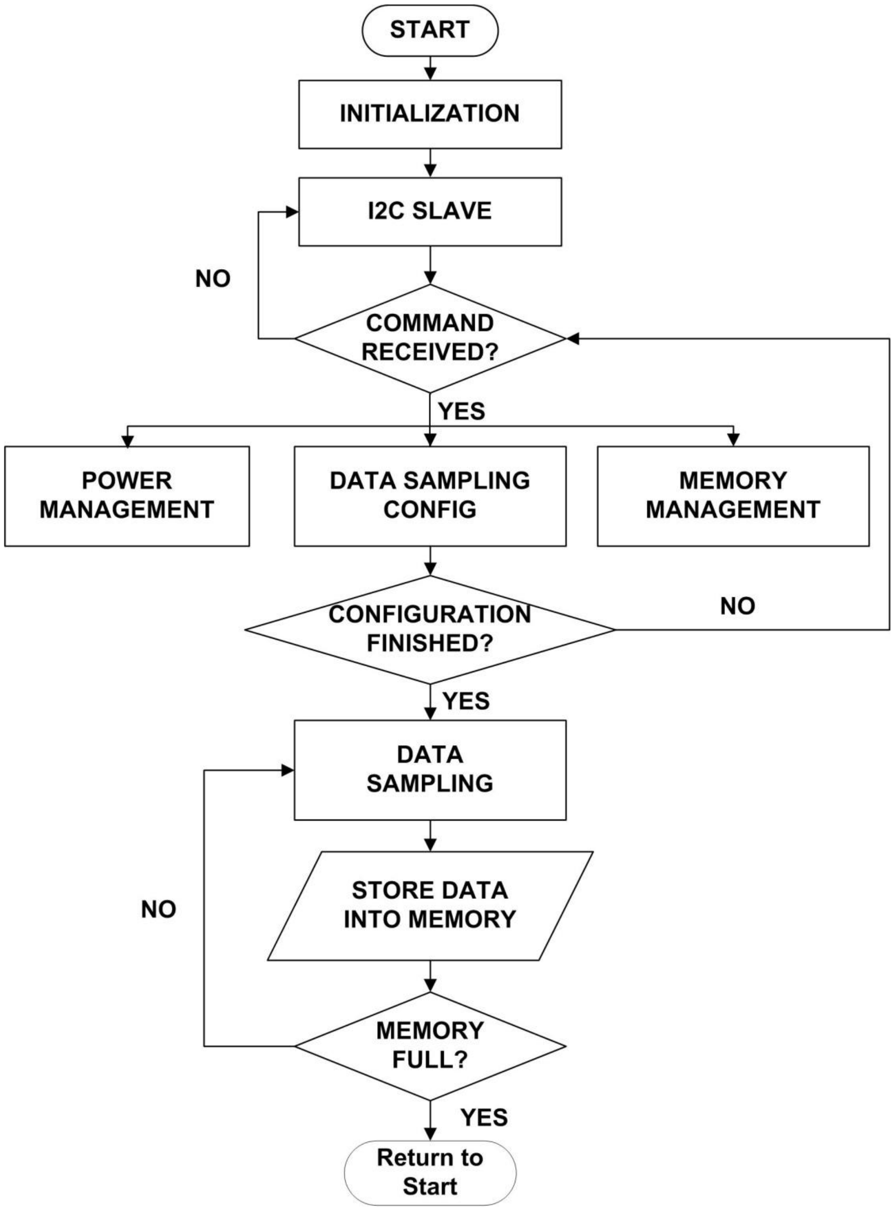


Figure 3.2. Overall structure of the BIRD sensor program

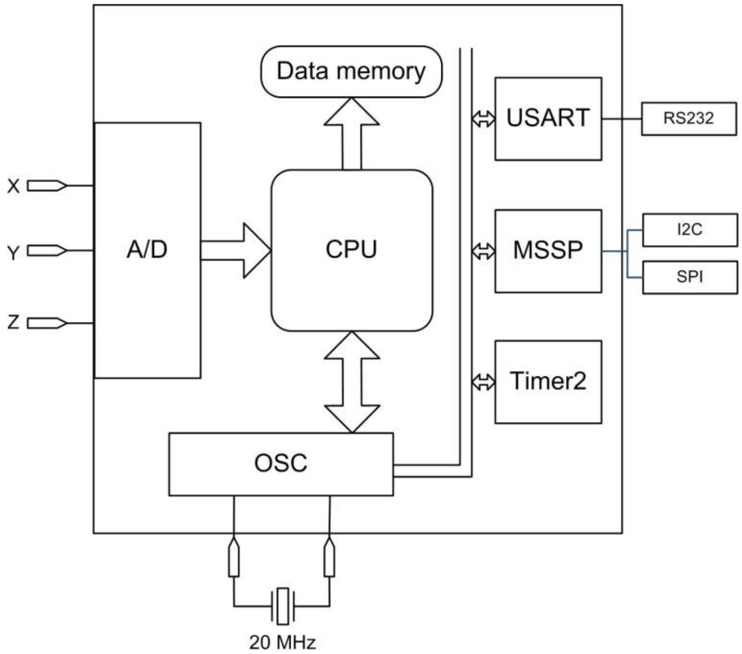


Figure 3.3. Diagram of the MCU hardware that related to the BIRD sensor program (adapted and modified from the PIC18LF2520 datasheet(Microchip))

different power modes is achieved by executing the corresponding commands. The acceleration sampling threshold has seven options(0 g, 18 g, 24 g, 41 g, 53 g ,99 g and 205 g), which covers a range from the low g level to medium g level. This range provides enough options for filtering impacts at different levels that required in the field. The sampling frequency also provides five options (682 Hz, 998 Hz, 1480 Hz, 2210 Hz, and 3050 Hz). This sampling frequency range allows the sensor to record impacts the sensor experiences. The sensor' F-RAM can also be erased with one command. Clock synchronization is achieved by updating the timer of the microcontroller upon receiving real-time information from the computer.

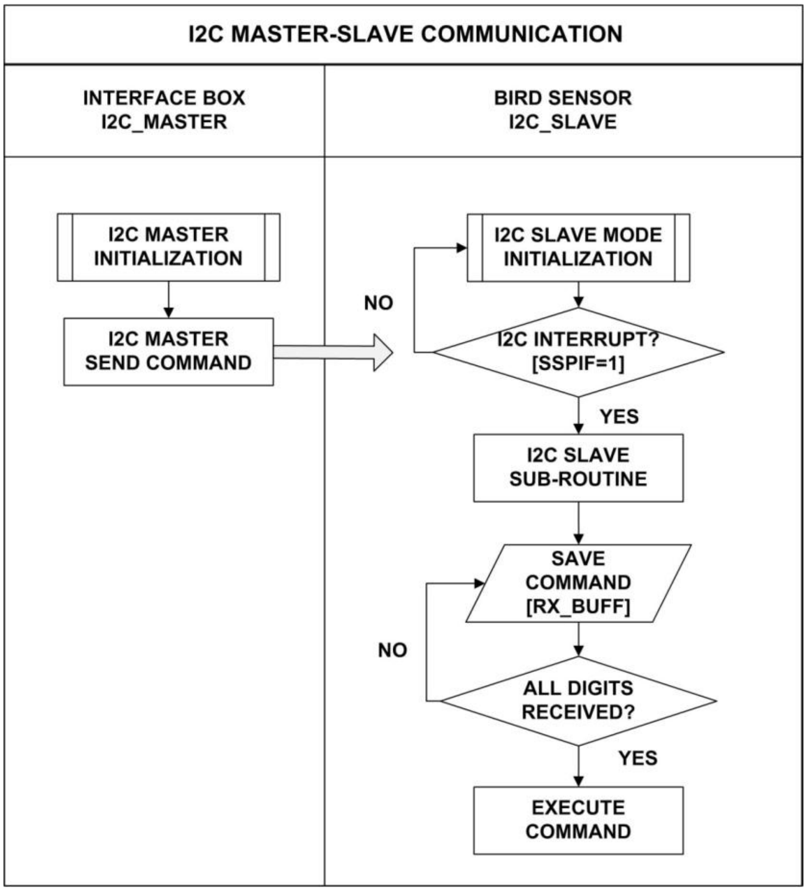


Figure 3.4. I2C Master-Slave communication between the BIRD sensor and the interface box

3.4.1.2 Data storage and memory management

The acceleration data are saved into the memory in a given format. As shown in Figure 3.5, each dataset has 13 bytes including the accelerometer data from the X, Y, and Z axes, the real time, and the impact index. Each axis of accelerometer data needs two bytes due to 10 bits A/D conversion resolution. The occurring time of each impact needs to be recorded in order to characterize the mechanical harvest process. To provide adequate timer accuracy for a maximum sampling frequency of 3.0 KHz, the resolution of the timer should be at least 1/3000 (s). The clock of the sensor was created with a resolution of 0.25 ms (TICK), using the interrupt service routine (ISR) based on the Timer 2 module that provided by the PIC18LF2520 microcontroller. The second is updated when the TICK equals to 4000. The hour (HH), minute (MM), and second

(SS) occupy one byte each, while the TICK uses two bytes. The clock's error introduced by the ISR was calibrated, which is discussed in later section. The impact index can count datasets up to 65536 with two bytes, which is adequate for a typical field test.

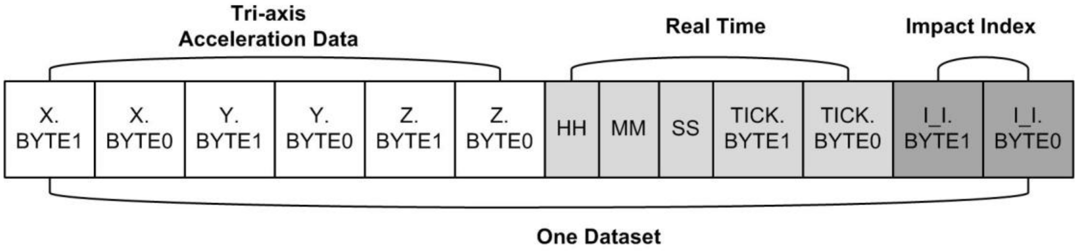


Figure 3.5. Data structure of each acceleration data set

Data are managed using four subroutines which can perform write, read, erase and upload. Data are written into the F-RAM memory chip using the hardware SPI protocol, as illustrated in Figure 3.6. The SPI module transfers the data serially with one byte each time. The thirteen bytes (one data set) are serially written into sixteen bytes of memory, leaving the last three bytes empty. The reason is the MCU may have address handling errors if the address is increasing continuously. During the data downloading process, the MCU on the BIRD sensor needs to read data from the F-RAM (SPI communication) on the sensor and write them into the F-RAM (I2C communication) on the interface box's I2C FRAM. Both the I2C and SPI communication protocol are established based on the MSSP module, so the MCU needs to force the MSSP module to either I2C or SPI mode before it communicates with the respective memory. Two subroutines were created to setup the MCU to either I2C or SPI mode. The corresponding subroutine needs be called when either mode is selected. The SPI writes data based on the bus speed of the MCU (5 MIPS), which ensures the MCU can record data in a fast pace (<=1.5 ms/impact) during the data sampling cycle.

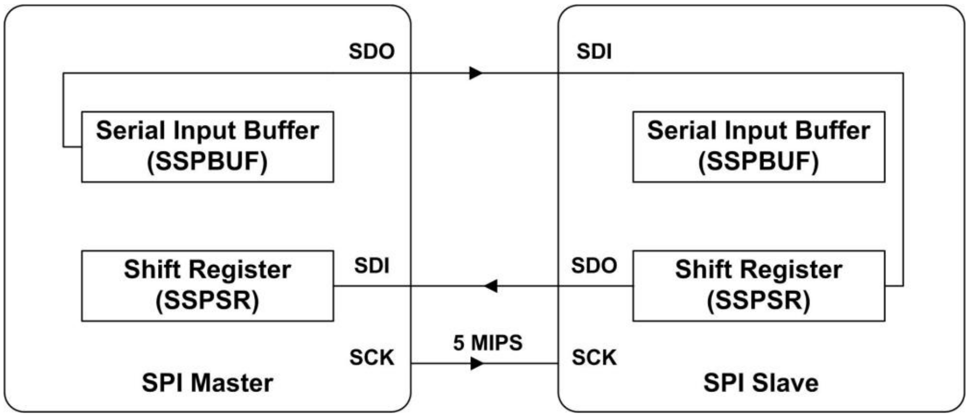


Figure 3.6. Hardware SPI Module based on the MCU's MSSP module

3.4.1.3 Acceleration data sampling and recording

The BIRD sensor records data based on the actual impacts it experiences. If sampled at a sufficient frequency, each impact can be depicted as a bell shape curve with certain time of duration, as shown in Figure 3.7. The vector summation is calculated based on three axes of acceleration data; the peak G is the maximum value of the vector summation. The area under this bell curve, calculated by the integration of the impact curve with impact duration, is defined as the velocity change. The velocity change is another important index to define an impact's aggressiveness. The threshold acceleration is selected to record impacts that are higher than it in order to avoid recording minor impacts which may fill the memory quickly. In order to record the whole shape of the impact to more accurately calculate the velocity change, certain data points under the threshold are also recorded: three before the impact data and six after the impact curve, defined as leaders and trailers, respectively. The three sections of acceleration data (leaders, impacts, and trailers) are recorded into different arrays but belong to the same impact. The LT_F flag is used to sort out impacts points that may belong to any of the three sections.

Figure 3.8 illustrates a whole sampling cycle for recording a complete impact: it starts with A/D conversion of analog accelerometer voltage into digital values. After converting the three axis's analog signals into digital values, the scalar value of the vector sum is calculated following equation 1. The sum is compared to the user selected threshold value. Impacts with peak G higher than the threshold will be written into the memory. As the threshold value has resolution of 5.86 g/count. This may allow the sensor to record impacts with peak G slightly lower than the threshold.

$$\text{Sum} = \sqrt{X^2 + Y^2 + Z^2} \qquad (1)$$

The total sampling cycle which records the leaders, impact data and the trailer determines the maximum sampling frequency, which can be up to 3.0 KHz. Figure 3.9 provides the breakdown analysis for the time used to record one data set. The AD conversion of three axes of acceleration values takes 60 µS in total, as defined by the AD conversion procedure in the microcontroller. It takes 268 µs for calculation and comparison of summation values, array update, and the timer

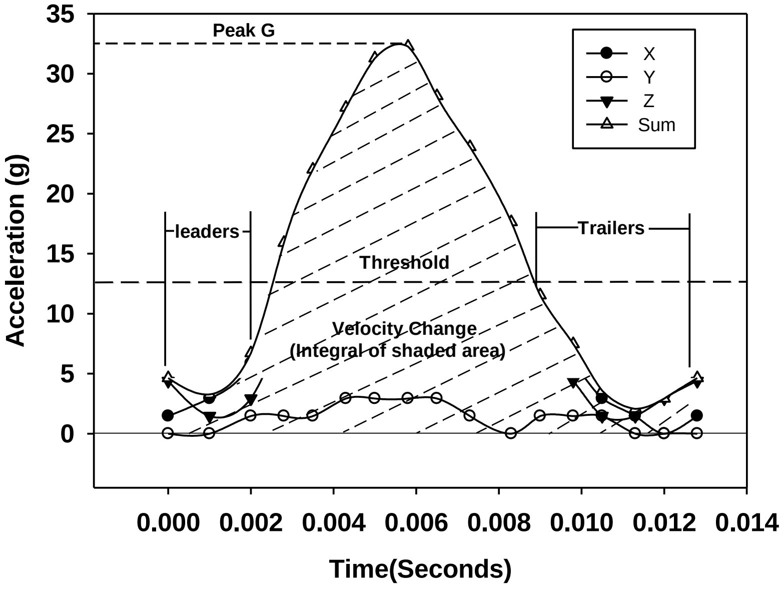


Figure 3.7. Impact curve recorded by the BIRD sensor

interrupt, determined by instructional cycles of each command. The void operations, which idles the system clock for given number of instructional cycles, are used to delay the sampling process for creating lower sampling frequencies. The sampling cycle length can be adjusted by change the number of void operations (N) with the maximum frequency of 3.0 KHz, when N equals to 0. Other four lower frequencies (682 Hz, 998 Hz, 1480 Hz, and 2210 Hz) were created by using the method of trial and error. All frequency options were verified by calculating the number of data points sampled within certain duration of time.

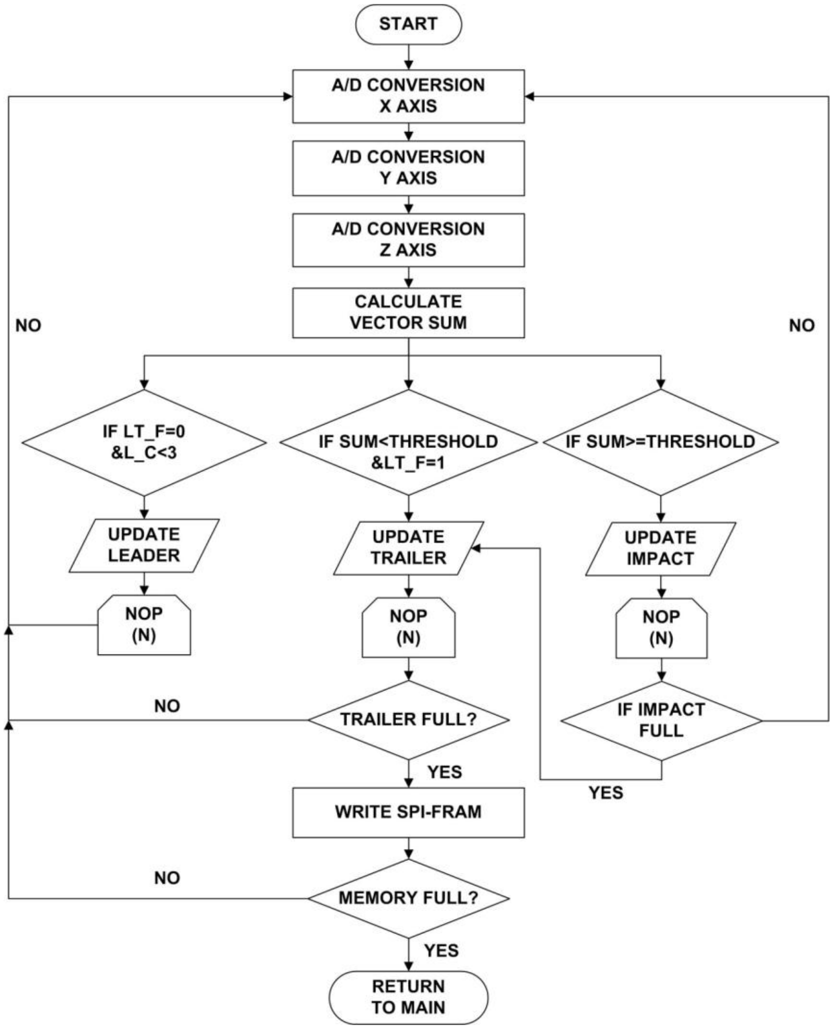


Figure 3.8. Flow chart of the impact data sampling cycle

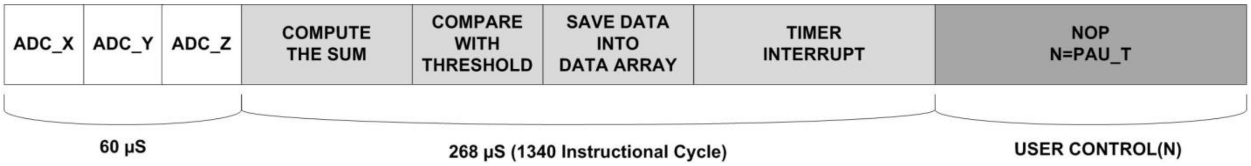


Figure 3.9. Duration of one sampling cycle

3.4.2 The interface box program

The program on the interface box establishes communication between the BIRD sensor and the computer. It transfers the control commands from computer to the sensor, uploads data from the sensor to the computer, and updates the informaiton on the LCD, as shown in Figure 3.10. The program on the interface box also monitors the battery level of the BIRD sensor by sampling a A/D channel connected with the sensor battery. The microcontroler on the interface box works in the I2C master mode when it communicates with the BIRD sensor. It receives commands and sends data to the computer via the RS232 serial communication. The interface box recharges the battery on the BIRD sensor when the battery votlage drops below the cuting off votlage (3.6 V). The user can turn on a switch for recharging the battery.

The program receives commands from the computer and executes that command’s request. Totally there are 9 commands sent by the computer. These commands are provided in the appendix C. Commands about sensor configuration are passed down to the sensor MCU via the I2C bus. The other commands management are executed for the data management on the interface box. Data is retrieved from the interface memory via I2C communication and uploaded to the computer via the RS232 communication The interface memory can be erased after data are uploaded.

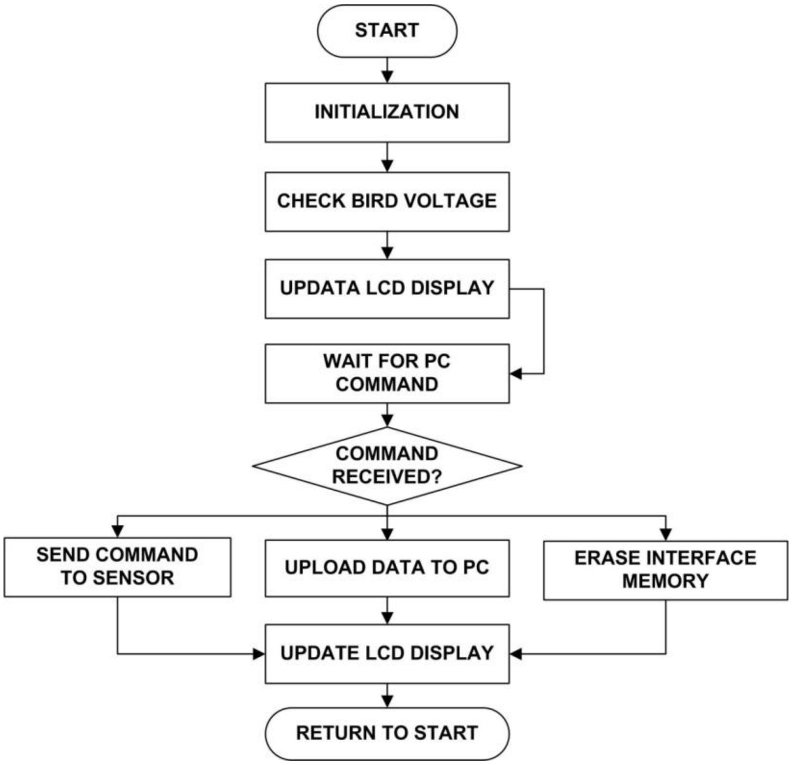
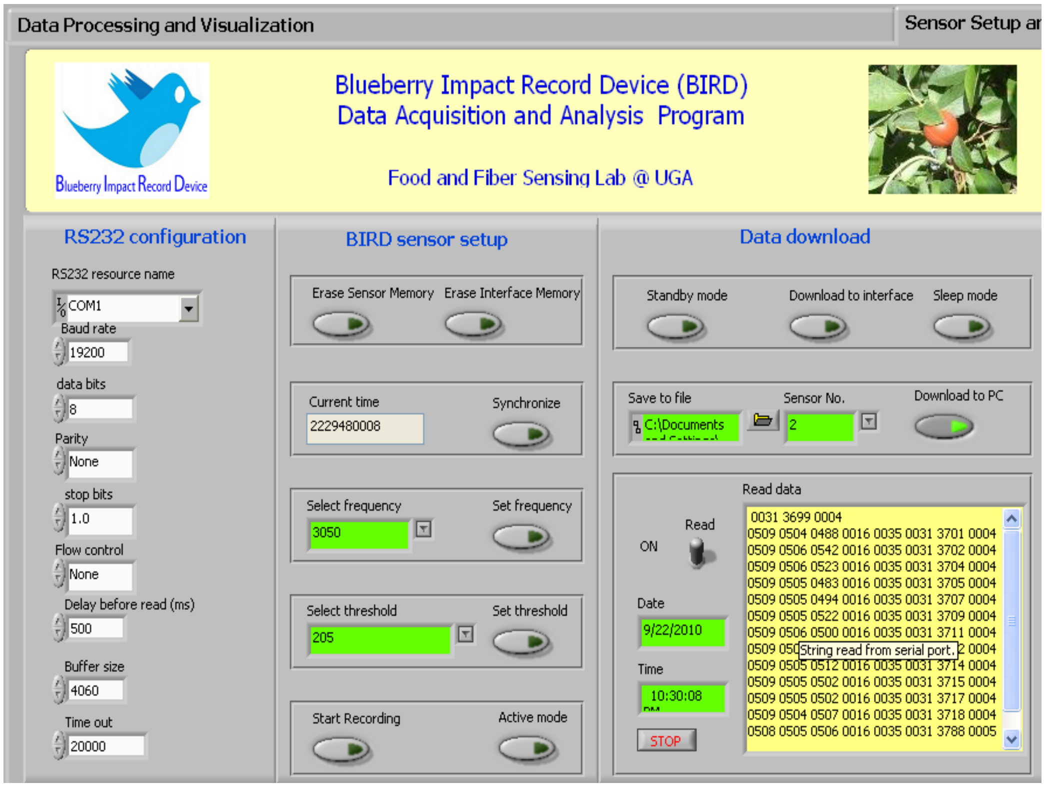


Figure 3.10. BIRD interface program structure

3.4.3 The i-BIRD computer software

The computer software i-BIRD was designed to have three major functions: communication with the BIRD sensor, data processing and display, and generating data reports. As shown in Figure 3.11, the "Communication and Data Download" interface has three sections: the RS232 serial communication setup between the i-BIRD and the interface box, the configuration of the BIRD sensor, and the data downloading and saving. Users can setup the RS232 communication through the VISA control in the LabVIEW. To setup the BIRD sensor, users need to erase both the sensor memory and the interface memory, and then synchronize the time between the computer and the BIRD sensor. Different sampling frequencies and threshold values can be selected based on users' needs. The power mode needs to be switched to "active" mode to enable the power supply to all components before starting data recording.

The third function of this interface is to manage the data downloading process. The power mode can be switched to the "standby" mode in order to save the power of the battery. Data are firstly downloaded from the BIRD sensor to the interface box before they are transferred to the computer and saved as ASCII files. A header file will be automatically generated when the data downloading occurs, which includes the synchronization time, sampling frequency and threshold, sensor number, and the file path. Downloaded data can be viewed through the "Read data" window in real time.



RS232 resource name: COM1
Baud rate: 19200
data bits: 8
Parity: None
stop bits: 1.0
Flow control: None
Delay before read (ms): 500
Buffer size: 4060
Time out: 20000

Current time: 2229480008
Select frequency: 3050
Select threshold: 205

Save to file: C:\Documents
Sensor No.: 2
Date: 9/22/2010
Time: 10:30:08

Read data

```
 0031 3699 0004
0509 0504 0488 0016 0035 0031 3701 0004
0509 0506 0542 0016 0035 0031 3702 0004
0509 0506 0523 0016 0035 0031 3704 0004
0509 0505 0483 0016 0035 0031 3705 0004
0509 0505 0494 0016 0035 0031 3707 0004
0509 0505 0522 0016 0035 0031 3709 0004
0509 0506 0500 0016 0035 0031 3711 0004
0509 050String read from serial port.2 0004
0509 0505 0512 0016 0035 0031 3714 0004
0509 0505 0502 0016 0035 0031 3715 0004
0509 0505 0502 0016 0035 0031 3717 0004
0509 0504 0507 0016 0035 0031 3718 0004
0508 0505 0506 0016 0035 0031 3788 0005
```

Figure 3.11. Communication and data download interface of the i-BIRD program

Data processing interface is shown in Figure 3.12, which processes the raw data and saves the processed data into spreadsheets.

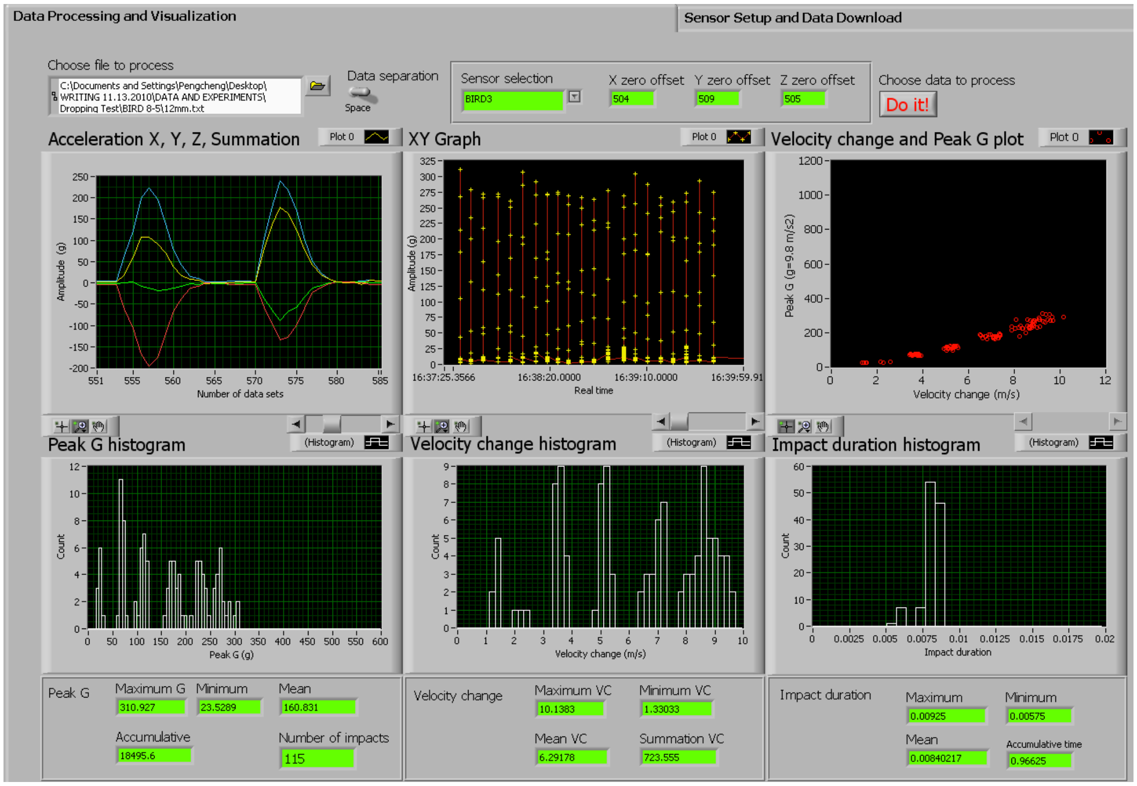
The data being processed can be displayed graphically for users' quick diagnosis and evaluation of the data. The three graphs in the first row describe the data from three aspects:

- Raw acceleration data without time. Impact data from the three axes of accelerometers and their summations are plotted in the first graph. This figure helps to interpret what are the raw acceleration values recorded by each axis and shape of each impact curve without time information. It also helps to locate each single data point based on the impact index. For instance, it can track acceleration values generated by the high speed spinning of the BIRD sensor, which are not actual dynamic impacts.
- Acceleration data with time.

  The second graph shows the impact curves against time. Each sampling point is a summation of three axes with the exact time provided by the clock on the BIRD sensor. The distribution of all single impacts can be identified. From the overall view, groups of impacts can be identified and separated with time information, corresponding to actual impact events recorded by the user. By zooming in this Figure, the real impact curve shape can be viewed, the impact duration, and peak-G value can be accessed. During the field test operation, it can also be used to identify experimental replicates, or eliminate irrelative impacts.
- Peak G against velocity change graph.

  The program integrates the area under each impact curve as the velocity change, The velocity change of each impact curve and its corresponding peak G value are plotted in one graph. Locations of impacts on this graph indicate their intensity which can be used to predicate the probability of creating the bruising.

The three graphs in the second row show histograms of Peak G, velocity change, and the duration of each impact. Descriptive statistical results of these parameters are also calculated and displayed.


Choose file to process: C:\Documents and Settings\Pengcheng\Desktop\WRITING 11.13.2010\DATA AND EXPERIMENTS\Dropping Test\BIRD 8-5\12mm.txt

Data separation: Space

Sensor selection: BIRD3

X zero offset: 504
Y zero offset: 509
Z zero offset: 505

Peak G
Maximum G: 310.927
Minimum: 23.5289
Mean: 160.831
Accumulative: 18495.6
Number of impacts: 115

Velocity change
Maximum VC: 10.1383
Minimum VC: 1.33033
Mean VC: 6.29178
Summation VC: 723.555

Impact duration
Maximum: 0.00925
Minimum: 0.00575
Mean: 0.00840217
Accumulative time: 0.96625

Figure 3.12. Data processing and display interfaces of the i-BIIRD program

3.5. Testing of the software

The software of the BIRD system was tested and evaluated in the following aspects: timer accuracy, sampling rate, impact curve shape, program speed and efficiency.

3.5.1 Timer accuracy analysis

The real time clock was created by using the interrupt service routine and Timer2 module of the microcontroller (PIC18LF2520). Error can be introduced from both the software and the resonator (CSTCE20M0V53Z-R0, Murata Electronics, Kyoto, Japan). The resonator has initial frequency error of 0.5%. Temperature drifting can also introduce error to the resonator speed, with 0.15% temperature stability within -40~120 ℃. Therefore, the software timer was tested and calibrated to ensure its best performance.

As shown in Figure 3.13, the BIRD sensor was mounted on the impact table with the Z axis being the sensing axis along the impact direction (up and down). The impact table has two vertically mounted sliding tracks. The impact board can be released from a given height and impact with the base. Acceleration values sampled from the Z axis were recorded by both the BIRD sensor and the NI-DAQ data logger (NI-6008, National Instruments, and Austin, Texas). The NI-DAQ data logger has higher sampling frequency and accuracy (11 bits input resolution and 14.7 mV absolute accuracy) and therefore was used to compare with the data collected from the BIRD sensor. The SignalExpress (Signal Express 3.0, National Instruments, and Austin, Texas) accompanied with the NI-DAQ was used to setup the data logger with 10 KHz sampling frequency using the Referenced Single Ended (RSE) signal mode. The computer time that NI-DAQ recorded was regarded as the standard time (resolution: 41.67 ns).

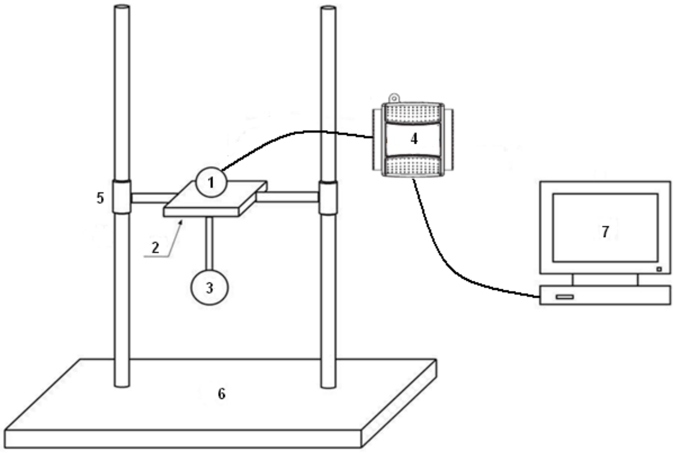


Figure 3.13. Impact curve calibration using impact table and NI-DAQ data logger Impact table: 1. BIRD sensor 2 Impact Board 3 Impact material (rubber ball) 4 NI-DAQ (Voltage Data logger, 12bit A/D, National Instruments) 5 Sliding Tracks 6 Base 7 LabVIEW Signal Express

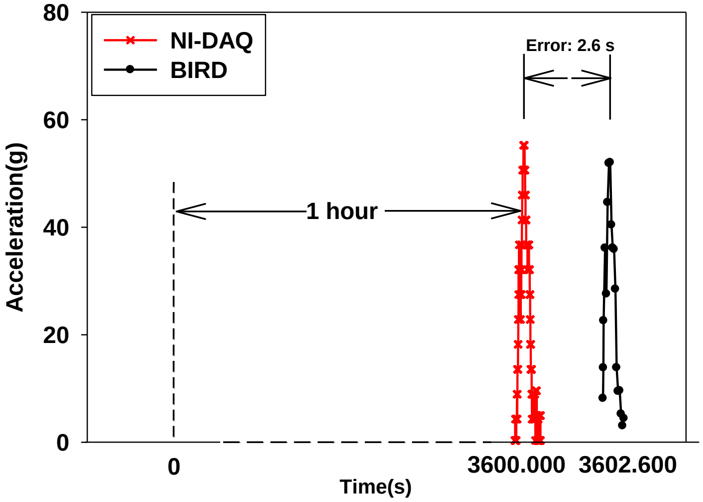


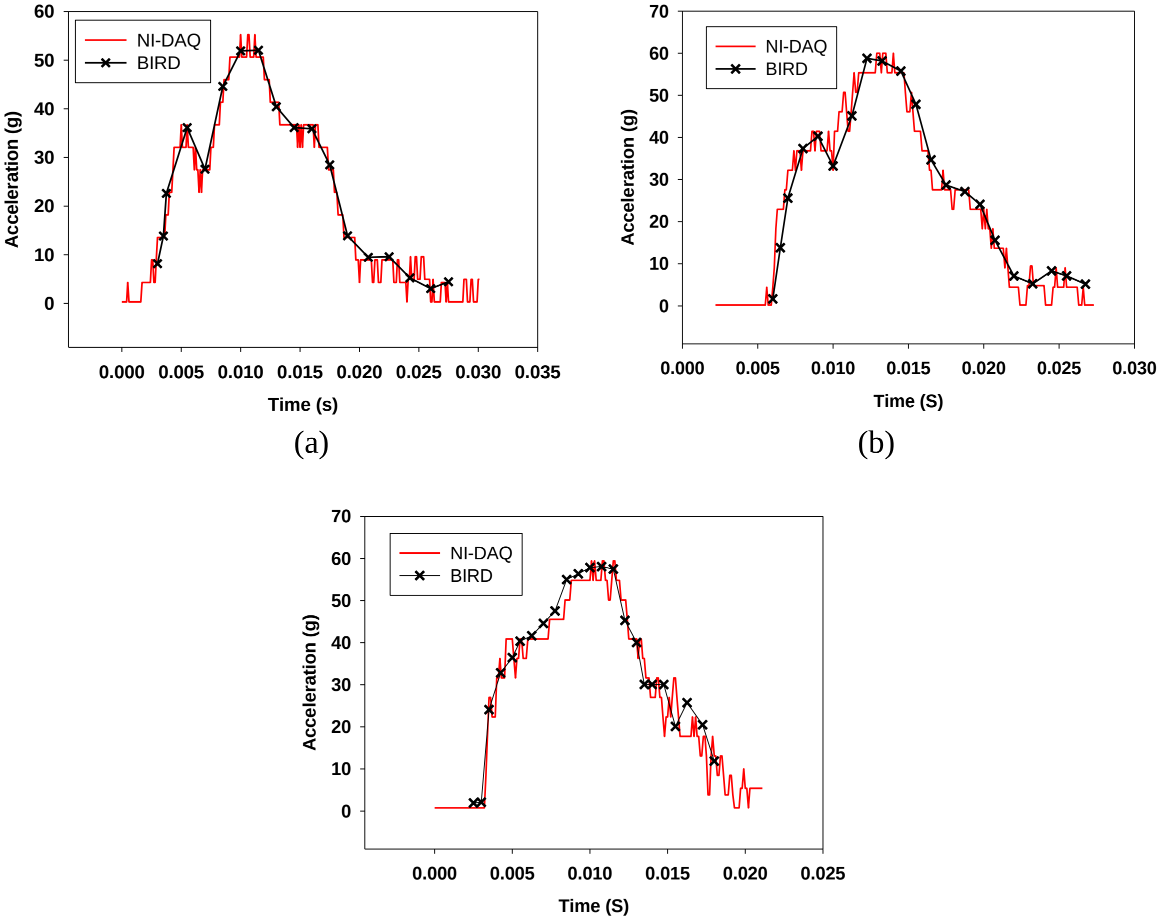
Figure 3.14. BIRD clock error calibration based on dynamic impacts

To calibrate the timer of the BIRD sensor, as illustrated in Figure 3.14, the BIRD sensor is synchronized with the computer time at time 0; the impact board was released every five minutes to generate impacts, which were recorded by both the NI-DAQ and the BIRD sensor. For the same impact, the difference between the computer clock and the BIRD sensor clock is regarded as the error of the BIRD sensor clock. The clock error of the sensor was calibrated at one hour, as the sensor clock usually can be synchronized again within one hour in the field. The minimum interrupt duration of the sensor was designed to be 0.25 ms (Tick). Therefore, the second should be updated every 4000 Ticks theoretically, the minute should be updated every 60 seconds, and the hour updated every 60 minutes. Three replicates were performed to calculate the average error of the timer. The tick was adjusted to 3998 after calibration, with the average error of 0.073%, which equals about 2.6 seconds' error in a whole hour. This accuracy is adequate for this application. The performance was also confirmed in impact curve test, which is discussed in the following section.

3.5.2 Impact curve shape and sampling frequency

The sampling frequency of the BIRD sensor can be confirmed by checking the number of data points recorded within certain duration of time. Five frequencies were confirmed using this method: 682 Hz, 998 Hz, 1480 Hz, 2210 Hz, and 3050 Hz. The maximum sampling frequency is 3050 Hz. In order to evaluate the distortion of the recorded impact curve, the sample impact curve recorded by the BIRD need to be evaluated. The experimental setup in Figure 3.13 was also used to generate dynamic impacts, the signal response of the accelerometers were both sampled by BIRD sampling software and the NI-DAQ. Three lower frequencies of the BIRD sensor (682Hz, 998Hz and 1480Hz) were selected to record the impacts, with ten replicates for each frequency, since higher sampling frequency can record impacts with better accuracy. The impact curve shape recorded by BIRD sensor and the NI-DAQ matched well with each other in all ten replicates for the three selected sampling frequencies. A typical impact curve among the

10 replicates is presented in Figure 3.15 for each frequency. The sampling software can record exactly about the impact, in aspects of impact duration, acceleration values and curve detail. The velocity change was also calculated, with relative error of 0.8%, 4.4% and 2.1% respectively. Previous successful instrumented spheres, like IS100, reported error of 5% for its peak G under dynamic test, the PMS60 reported an error close to 10%, while both sensors have their velocity change value not discussed. Since error of velocity change depends on G values and timer accuracy, the reported three velocity change errors can be acceptable (<5%). The errors didn't show linear along with sampling frequency since all frequencies' impact were generated differently from each other, error can be introduced from the resolution difference of the two sampling devices, the impact table's consistency and manual operation error when the table is released from a given height. The variations of error can be introduced from the noise of the NI-DAQ itself, the impact table's consistency and the distortion introduced by sampling frequency. Due to the limitation of the data array size which is 29, a proper sampling frequency need to be selected to ensure the whole impact curve is recorded in the data array. The duration of impact is can be calculated through the number of impact points divided by the sampling frequency. Increasing sampling frequency will reduce the allowable maximum duration of a complete impact curve. The impact curves listed in Figure 14 (~20 ms) are much longer than the sensor experiences in the field, which typically is around 6 ms.



(a) (b)

(c)

Figure 3.15. Comparison of impact curves recorded by BIRD sensor and NI-DAQ data logger (a) 668 Hz (b) 998 Hz (c)1480 Hz The impacts created for the three different frequencies are different, in order to test the sampling software's accuracy in retain impact curve shape under different situations. Low frequency of the BIRD can still retain low impact curve distortion by comparing with the high frequency sampled impact curves that recorded by NI-DAQ 6008

3.5.3 Speed and efficiency

The SPI bus for sensor memory also uses the same bus speed (5 MIPS) as the MCU. The I2C communication between sensor and interface board uses I2C Master Slave mode, which transmits data at 400 KHz bus speed. Communication between interface board and the computer software uses RS232 at Baud rate 19200. To evaluate the system's speed performance to guide through the field application, each operation procedure was clocked to test its performance. Test of erase sensor memory, upload sensor data to interface box and upload sensor data to computer relies on the interface box LCD to indicate the start of finish of that operation. Write an impact array to the SPI memory on the sensor was estimated based on the programming commands used.

The speed for erasing sensor memory is much faster than the upload sensor data to interface board and upload data to the computer since only the SPI protocol is used in the former. The I2C and RS232 communication slow down the overall data download process, however, those time required are not significant during field operation, based on the sensor's battery performance(3 hours) and memory size(~500 impacts), which usually requires half hour between downloads. Table 3.1 listed the important operations of the BIRD software system and their performance.

Table 3.1 BIRD operation speed lists

Operation	Protocol	Maximum time(S)
Erase the sensor Memory	SPI	6
Upload sensor data to interface board	SPI,I2C	35
Upload to PC	RS232	68
Write an impact array into SPI memory	SPI	0.0015

3.6 Conclusion

The BIRD software reached initial design criteria both in functionality and performance. It can record three axes of acceleration data based on a single impact. The sampling rate can reach 3.0 KHz maximally, the dynamic impacts recorded by the sensor was verified by comparison with the same impact sampled by high frequency signals from the NI-DAQ data logger. The interface program can establish the communication between sensor and the computer. The i-BIRD software was designed and applied for data downloading and analysis. The software's operation speed was also evaluated for guiding actual applications. The BIRD software is a complete data acquisition, processing and analyzing platform.

CHAPTER 4

BLUEBERRY MECHANICAL HARVEST BRUISING STUDY USING A MINIATURE BLUEBERRY IMPACT RECORDING DEVICE (BIRD)

4.1 Overview

Susceptibility of blueberries to impact damage can be attributed to factors such as cultivar, harvest and handling techniques. Despite breeders' effort in breeding new cultivars to increase the damage tolerance of blueberries, , bruising damage that occurs during the mechanical harvest process still prevents the mechanically harvested blueberries from going to the fresh market. Blueberry dynamic experiences during the mechanical harvest process are quantifiable but there has been no previous research on this subject. This study was conducted using a miniature Blueberry Impact Recording Device (BIRD) with three objectives: first, a sensor was used to quantify the dynamic impacts which occured during the mechanical harvest process. Second, a sensor was used to evaluate the rotary, slapper and sway harvester's overall performance. Finally, the sensor's impact data was used to predict a single impact's bruising probability based on the correlation between sensor data and the actual bruising probability of blueberry. To quantify the mechanical harvest process, the rotary harvester's field operation was simultaneously recorded by close-up video and tested by the BIRD sensor, with four major phases of the sensor's dynamic experiences identified. The impacts recorded in phase one (beaters and tunnel), phase two (fish scales and conveyor belts), and phase four (conveyor belt to lugs) accounted for 20.5%, 43% and 24.3% of the total accumulative peak G, respectively. No dynamic impact was found in phase three (conveyor belt transfer process). For the three harvesters' overall evaluation, firstly, the mounting height of the sensor and beater speeds were analyzed about their influence on the impacts collected for each harvester. In the overall evaluation, the sway, slapper and rotary harvesters generated average impact numbers of 34.8, 32.8, and 20.2, maximum peak G of 473.9 g, 428.2 g and 350.9 g, average peak G of 103.6 g, 97.9 g and 90.1 g and the accumulative peak G of 3540.5 g, 3168.1 and 1755.2 g, respectively. The rotary harvester was found to be the least violent harvester according to all four parameters. The harvester surface materials were compared, including the fish scale, the steel tunnel and the conveyor belt. The peak G of the steel conveyor belt (slapper, sway) was two times of that of the plastic conveyor belt (rotary).The sensor's impact data was correlated with the bruising probability of three cultivars. With all those successful tests, the sensor can be a useful tool for evaluating and reducing bruising damage of the blueberries in the mechanical harvest.

4.2 Introduction

With a total yield of 449.8 million lbs of blueberries in 2009, half of the US blueberry production went to the fresh market (Economic Research Service, USDA). The majority of these blueberries were still harvested by hand due to the low quality of the blueberry fruits that were mechanically harvested. The increasing high labor cost and low harvest efficiency created a bottle neck for the further development of blueberry industry (Brown, D.E.Marshall et al. 1983).

Using mechanical harvesters to replace traditional hand harvest is the key to improve the overall efficiency. However, a serious existing limitation is the bruising damage caused by impacts. Blueberries are exposed to frequent impact damages during the interactive mechanical harvest process with the majority (78%) of the them were bruised (G.K. Brown 1996). The impact damage is preventing the mechanical harvesters for fresh market use.

There are three major mechanisms to detach the ripe blueberries from bushes (Mainland 1993) : The first one is to use slapper rods that extend from a vertical axis to hit the bush from both sides (slapper machine). Another way is to sway rods moving in the same directions to compress the bush moving side to side (sway machine). The last one consists of horizontal fingers extend from a vertical spindle (rotary machine) that can roll through the bush with vertical or horizontal vibrations. A recently improved V45 experimental harvester has been proven to have some advantages in reducing mechanical damage, which can be attributed to the short falling distance of blueberry fruits, cushioned catching surfaces and less collisions between blueberries and bushes or harvesters (Takeda, Krewer et al. 2008). Though it proved that the V45 machine had advantages in producing higher quality blueberries with less bruising and internal damage, it caused much more limb damage especially for the bushes that were not pruned. Methods employed by the advanced harvesters were similar: reducing the impact forces on the blueberries during the harvest. However, how the blueberries were bruised within the mechanical harvesters was unknown. A quantitative study about locations of impacts and magnitude of the impacts is essential to improve current blueberry mechanical harvesters.

Many studies have been done on measuring bruising damage for large fruit and vegetables by using instrumented spheres. Essentially, an instrumented sphere is a data logger that is subjected to the same loads as real fruits and vegetables. The load experiences are recorded using either acceleration data or pressure data. Sensors that record acceleration values can output data as single impacts. The maximum acceleration value of an impact is the peak G and the integration of the impact’s acceleration values against time is its velocity change. Previous literature well stated how the Peak G and Velocity Change(VC) can be combined together to reveal a single impact's severity (Schulte, Brown et al. 1992). Products like PMS60 (Herold, Truppel et al. 1996), Impact Recording Device (IRD, Techmark MI), and the PTR100, with its later version PTR 200 (Canneyt, Tijskens et al. 2003) have been widely used for fruits and vegetables. Schulte and Brown performed drop tests for different varieties of apples and the IS100 was used to determine the impact conditions which initiated bruising on difference surfaces (Schulte, Brown et al. 1992; Schulte, Timm et al. 1994). The bruising thresholds were successfully developed using the IS100 response under the same drop conditions, which can be used to guide the improvement of apple packing lines. Mathew and Hyde dropped both potatoes and the IS100 to different surfaces at selected drop heights to identify the threshold peak G for the zero damage (Mathew and Hyde 1997). Damage threshold for four cultivars of peaches were also identified for steel surface using the IS (Lin and Brusewitz 1994; Schulte, Timm et al. 1994). The Pressure Measuring Sphere-60 (PMS-60) was used to identify the damage thresholds for onion (Herold, Oberbarnscheidt et al. 1998), and prediction of tomato stem-puncture injury (Desmet, Van linden et al. 2004) However, these sensors all have limitations for the blueberry. For instance, the most commonly used instrumented sphere, the Impact Recording Device, has a diameter of 57 mm (~ 2 inches) and weighs 96 grams, which is too big to go through the blueberry mechanical harvesters. Another constraint is that the sensor has limited drop height, which requires no more than 10 cm when being dropped to the hard surface. Further, this sensor has been mostly used for evaluation of packing lines, rather than the mechanical harvest process. The PMS-60 is an sphere sensor that uses a pressure sensor to record both dynamic and static loads (Herold, Truppel et al. 1996). The sensor weighs 180 grams and has a diameter of 62 mm, which is also too big for the blueberry use. It has maximum drop height of 100 cm, while actual application requires 150 cm or higher.   In addition to the size constraint, the PTR200's sampling frequency is only 100Hz, which would not be adequate to record the shape of the dynamic impact curves.

The miniature sensor that was specifically developed for blueberry bruising evaluation is a promising tool. It was aimed to simulate blueberries by being mounted on blueberry bushes and harvested by mechanical harvesters exactly as what a real blueberry experienced. Recorded sensor data were used to quantitatively describe the harvest process and to identify the critical points where bruising most likely happen. The BIRD was used for the first time to measure mechanical harvest process of blueberries, which has never been done before.

The three objectives of this research were to:

1. Quantitatively describe the mechanical harvest process of a typical rotary harvester.
2. Compare the performance of the rotary, slapper and sway mechanical harvesters and their surface materials.
3. Correlate the BIRD sensor data with bruising rate of blueberries.

4.3 Materials and methods

4.3.1 Blueberry Impact Recording Device (BIRD)

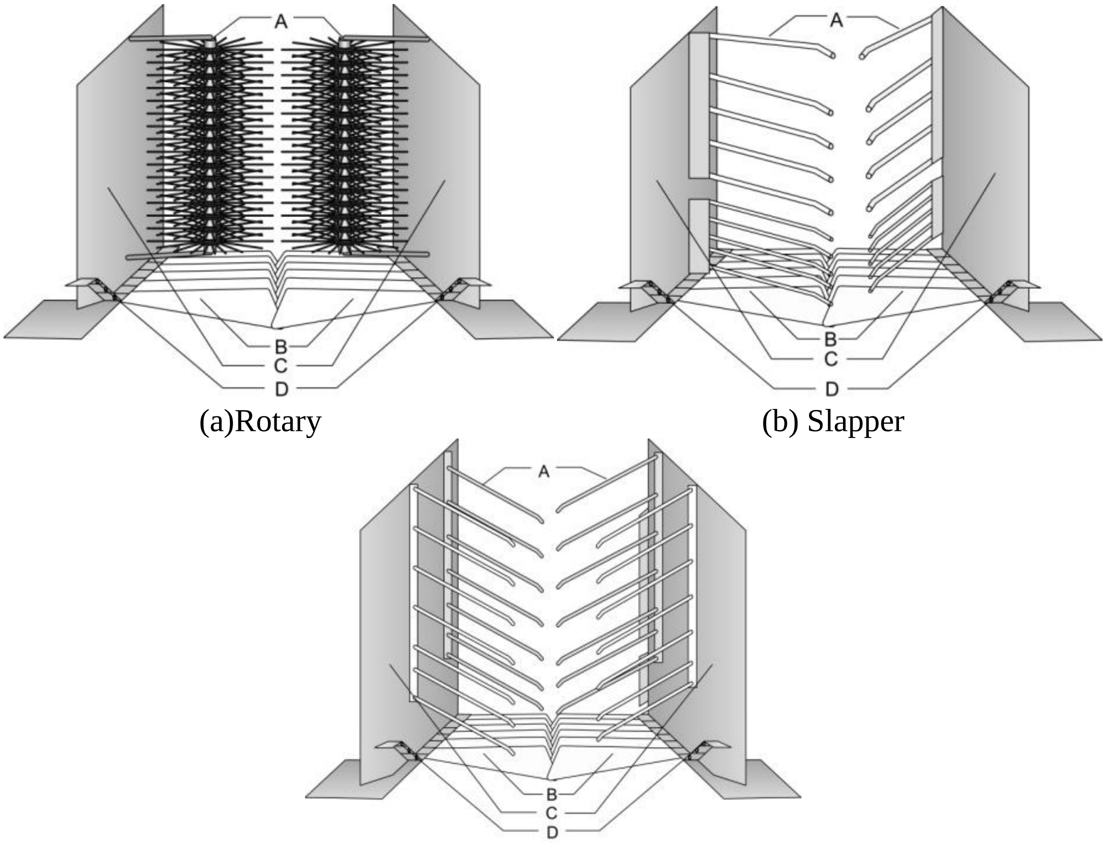
The BIRD system can record dynamic impacts with real time information. It includes a miniature impact sensor, an interface box connecting the computer with a serial cable, and the i-BIRD software installed on a personal computer. The BIRD sensor is a rubber sphere with a diameter of 25.4 mm (~1 inch) and weighs 14 grams. This miniature size allows it to go through the mechanical harvest process as a real blueberry fruit. The sensor collects three axes of acceleration values with dynamic range of ±500 g (g=9.8 m/s^2) and maximum sampling frequency of 3.0 KHz, which is fast enough to capture the impacts it experiences. Summation of three axes can reach 865 g. From the data collected in the field, less than 0.02% of impacts exceeded the sensing range. The BIRD sensor records data based on the actual impacts it experiences. Each impact data provides information about the raw data with time, the Peak G, and the velocity change. The sensor has a rechargeable battery that lasts for three hours and needs two and a half hours to be recharged. The interface box works as an intermediate interface to connect the sensor with the computer. It also has a double line-monochrome LCD to display the operation status. In addition, a computer software (i-BIRD) was designed in LabVIEW graphical programming language environment to configure the sensor, download and process the data, and graphically display the results. Users can set the senor with given options of sampling threshold and sampling frequency. The sensor's clock can also be synchronized with the computer time before data collection. The downloaded data can be processed and saved into spreadsheets as data report, which includes the raw acceleration data, the processed data and the Peak G_VC data. The results can also be graphically displayed for quick view of the results. Four BIRD sensors have been used during the field application to improve the efficiency of data collection. The performance of all sensors have been calibrated and tested under the same routine to keep their uniformity and accuracy.

4.3.2 Rotary, slapper and sway blueberry mechanical harvesters

Mechanical harvesters of blueberry can be categorized into three major types: rotary, slapper and sway. The three types of harvesters have four common parts (Figure 4.1): actuator, fish scale, tunnel, and conveyor belt.

The actuators (beaters) are the major difference among the harvesters. Rotary, as shown in Figure 4.1 (a), has two spindles of beaters with 18 disks on each spindle, where the plastic beating fingers are mounted. The beaters can vibrate but do not have proactive rotational speed.

With the harvester going forward, the two spindles of beaters can rotate only when being pushed against by the blueberry bushes. The adjustable vibration from the beaters shakes off ripe blueberries from the bushes. The slapper and the sway harvester have different designs, as shown in Figure 4.1(b) and Figure 4.1(c). The two rows of beaters on the slapper harvester swing from opposite directions like "slapping" on the bushes from both sides while the sway harvester's beaters swing follow the same direction, and "swaying" the bushes. Both "slapping" and "swaying" can reach violent speeds, which can hit the blueberries with severe impacts.



(a)Rotary (b) Slapper

(c) Sway

Figure 4.1. Three major types of mechanical harvesters; A: harvester beaters, B: fish scale, C: Side tunnel of harvester body, D: Conveyor belt

The actuators of the harvester can shake, slap or sway the bushes; blueberries are forced to detach and collected by the harvester's blueberry fruits catcher, which typically comprises two rows of fish scales and two conveyor belts. The fish scale has overlapped edges to minimize ground loss of the blueberry fruits: The two rows of fish scales mounted on each side of the tunnel can be separated by the trunk of bushes, and then closed after passing the bush; this design allows the harvester to go forward, without opening significant room at the bottom. Blueberries fell on those fish scales roll into the conveyor belts on both sides, and are transported to the rear of the harvester. Fish scales are made of hard plastic surface, the conveyor belt is also made of the plastic for the rotary harvester, but it is made of the steel for the slapper and sway harvesters. Since the fish scales and the conveyor belts cover the majority of the catcher surface, the blueberries' impact on them can't be avoided.

4.3.3 Video record of model bush test

In order to describe the dynamic impacts that the BIRD sensor experiences during the mechanical harvest process, a video camera (Canon T2i, Canon, Tokyo, Japan) was used to film the test process. The rotary harvester was selected as a model harvester and one BIRD sensor was used for this video test. The sensor's clock and the camera's clock were synchronized with computer time before each test. In the field, the sensor was mounted on the first and last bush of one row and the bushes were defoliated, in order to provide a better view of the sensor. Forty replicates were tested. Replicates with sensor's dynamic impacts that were clearly recorded by the video camera were selected to be synchronized with the sensor data. With 60 fps (16.7 ms resolution) video speed, the locations of the sensor's dynamic impacts can be identified. The video frame numbers were recorded to provide the real time for each impact (video's starting time plus the elapsed time of that frame). On the other side, the impacts recorded by the sensor can be pinpointed with its actual location from the video, based on the provided real time. The impacts identified at each location of the harvester provided a quantification of the harvest process.

4.3.4 Evaluation and comparison of rotary, slapper and sway harvesters

The complete harvest process for the rotary, slapper and sway harvesters were evaluated using the recorded impacts. The BIRD sensors were mounted on the bushes and they were left to go through a complete harvest process. Rabbiteye blueberry bushes were selected with similar bush height (6~8 ft), limb density and ground conditions. Depending on the length of the row, 8~10 bushes were selected, with 5 bushes' interval or more between two replicates. The interval allowed enough space for remounting the sensors on the following replicate. Four sensors were alternatively mounted on the bushes and harvested by the mechanical harvester. The harvester needed to make a full stop before the sensors were retrieved and remounted on the subsequent bushes. Replicates with sensors fell into the lug were counted as successful. The mounted sensors had frequent premature ground drop or stuck on bushes due to inexperienced mounting methods, such as using a needle to hook the sensor on the bush. This problem was resolved by using a three-leg sensor holder that can flexibly control the detaching force, as shown in Figure 4.2. The majority of the replicates were successful after this modification. Because repeated tests may incur damage to bush limbs, switching of rows was performed based on grower's recommendations. As blueberry fruits can locate at different locations of one bush, two mounting heights were selected. The high position was (H, 0.8 of the bush height) about 150 cm (~ 5ft) and the low position (L, 0.4 of the bush height) was around 60 cm (~2 ft). Minor adjustments were made for the slapper harvester's low mount position since a tire mounted at the head of the harvester can force the sensor to detach prematurely. Two beater speeds were also selected for the three types of harvesters to test their potential influence on the bruising damage. The slow speed (S) is selected with standard of early season harvest and the fast beater speed (F) is used for post harvest season, when all blueberry fruits need to be cleared. In total four treatments were tested (H-F, H-S, L-F, L-S), and each treatment had 20 successful replicates. Due to logistic difficulties, the H-F and L-F treatments for the sway harvester only had 17 and 4 replicates, respectively. The BIRD Sensor was configured with a sampling threshold of 24 g for all tests, since it only requires a minimal drop height to generate impacts with peak G higher than 24 g (< 5 cm based on preliminary tests). Another reason of using this threshold, as the resolution of the sensor's threshold was approximately 6 g; the zero offset (0 g) of the sensor can varies. Therefore, a slight higher threshold was essential to filter noises that can be potentially recorded as real impacts. From drop height of 15 cm (6 in) to a hard surface, previous tested blueberries have no berries rated as bruised based on the provided rating standard (bruising is less than 25% of total area)(G.K. Brown 1996) . The sampling frequency was set at 2.2 KHz or 3.0 KHz, both of which are adequate to retain the data accuracy. Later field tests used 3.0 KHz because the sensors' memory capacity was upgraded from 64 KB to 128 KB, which allows more impacts to be recorded.

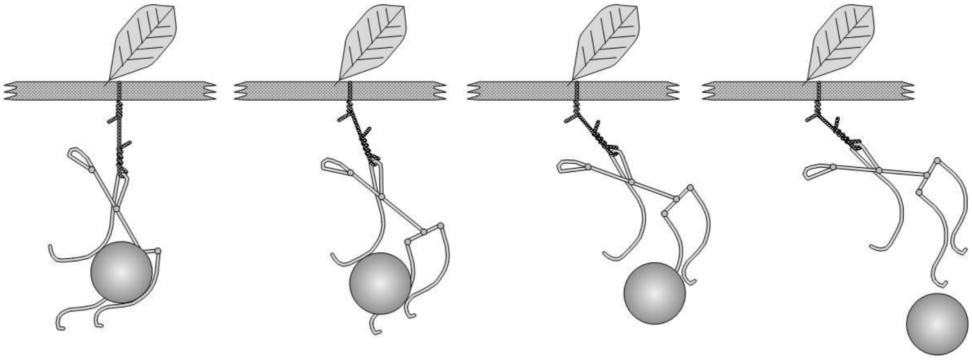


Figure 4.2. Sensor holder and sensor detach process

4.3.5 Harvester surface evaluation

The sensor interacted with the blueberry harvester mainly in the following three surfaces: the fish scale, the steel tunnel and the conveyor belt. The steel tunnels of all three harvesters were similar to each other, so only one of them (rotary) was evaluated. The fish scales of the sway and the slapper harvesters were identical to each other, so only one was tested to compare with rotary's fish scales. Conveyor belts installed on the slapper and sway are made of steel while the one on the rotary harvester was made of plastic. So, one steel conveyor belt was tested to compare with the plastic conveyor belt. In order to evaluate the surface difference of the three harvesters, the BIRD sensor was dropped to the fish scale, the conveyor belt, and the steel tunnel of the harvesters to compare their characteristics. Five drop heights were selected (15 cm, 30 cm, 60cm, 90 cm, and 120 cm), with 20 replicates for each drop level.

4.3.6 Correlating BIRD sensor data with three varieties of blueberry bruising

Both the BIRD sensor and blueberries were dropped on hard and soft surfaces. The fish scale was selected as the hard surface and one soft padding material (No Bruze, PORON Urethane Foam 4790-92-15250-04 6.35mm, Rogers Corporation, Woodstock, CT) was selected to provide contrast. For each material, the BIRD sensor was dropped at 5 different heights: 15 cm, 30 cm, 60cm, 90 cm and 120 cm. Three cultivars of blueberries (Scintilla, Farthing, and Sweet Crisp) were dropped to both surfaces. All blueberry samples were hand-harvested. Berries were picked gently and placed into plastic pails to minimize initial bruising. A control experiment with berries non-dropped was conducted to provide comparison with the dropped ones. The control berries were checked for bruising within 2 hours of hand harvesting. The rest samples were rated for bruise the following day. All samples were held at ~21 C ^{o} or room temperature. Dropping heights were 30 cm (~1 ft) and 60 cm (~ 2 ft). For each drop height, there were three replicates with 25 blueberries for each replicate. Blueberries were rated through the images of the sliced berry fruits along the equator area; bruising area that larger than 25% of sliced surface was rated as bruised. In order to compare different cultivars' bruising performances, another new cultivar 05-528 was also dropped to the fish scale surface, and the bruising rate for four cultivars were compared (Farthing, Scintilla, Sweet crisp, 05-528).

4.3.7 Data processing and analyzing

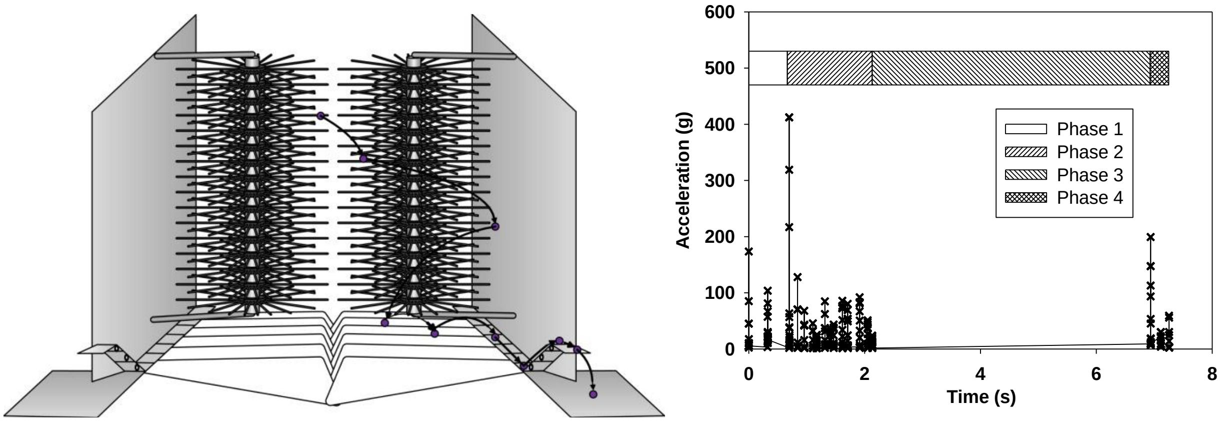
Data collected by the sensor were processed firstly using the i-BIRD computer software that was developed under LabVIEW environment. The raw data was separated into single replicates to be processed by i-BIRD. i-BIRD also generated sensor data reports, include a raw data file, a processed data file and the Peak G-VC file. The one-way analysis of variance (ANOVA) tests with Tukey's studentized range test were processed using the SAS (SAS Statistics, SAS, Cary, NC) to compare the test results. Comparison of harvester surface employed t-test ($P \leq 0.05$) for the peak G values.

4.4 Results and discussion

4.4.1 Quantitative description of the mechanical harvest process

The dynamic experiences of the BIRD sensor during the mechanical harvest process were recorded by a close-up video. The video and the data collected by the BIRD were synchronized to identify the location of each impact. Figure 4.3 (a) illustrates a typical experience of the BIRD sensor during a rotary harvester's harvest process. Figure 4.3 (b) provides the impact data recorded by the BIRD sensor with four distinct phases identified by the video recorder. The raw data's real time information was not used in this example for illustration purpose.

The phase 1 as indicated by the horizontal bar in Figure 4.3(b), is from 0~0.7 s, corresponding to the process between the detaching from the bush to the landing on the fish scale. In phase 1, the sensor may collide with beaters and the tunnel of the harvester, as well as bush limbs (Figure 4.3(a)). In phase 2 (0.7 ~2.2 s in Figure 4.3(b)), the sensor fell on the fish scale and rebounded to the conveyor belt with a set of impacts. The sensor's first contact with the fish scale was the severest impact (408 g, Figure 4.3(b)), due to the hard contact surface and the high falling height. Peak G of subsequent rebounds was reduced gradually. In phase 3 (2.2~6.9 s in Figure 4.3(b)), the sensor was transferred on the conveyor belt to the collecting lug. There was no dynamic impact generated during this process. In the last phase (6.9~7.3 s), the sensor fell to the blueberry collecting lugs from the conveyor belt, typically with several impacts on the plastic surface of the lug when the sensor was bouncing around. The first impact was harsh (200.2 g), since the sensor was dropped at a 35 cm height.



(a) (b)

Figure 4.3 Dynamic experience of BIRD sensor during rotary mechanical harvest process;(a) Physical contacts demonstration (b) A typical case of the real time impacts recorded by BIRD sensor.

Eight successful replicates of this video test were identified out of the total 40 replicates. The impacts' locations with the corresponding sensor data were summarized in Figure 4.4. The values labeled on each stack are the summation of the eight replicates. The average values were not calculated because impact with certain surface may not occur during some replicates.

The ratio of the accumulative peak G of each contact surface to the overall accumulative peak G was calculated and shown in the first bar of Figure 4.4. The result shows that about 67% (11362/16994) of the accumulative peak G was generated in phase 2 and phase 4. As observed from the video, the sensor's contact with fish scales, the conveyor belt and the lugs had several rebounds, while actual blueberries were less bouncy. This can be attributed to the difference in surface properties between the sensor and the actual blueberry fruits. Therefore, the rebounds on those three surfaces were removed, with result plotted in the second bar of Figure 4.4. Using this method, accumulative peak G of the impacts with the beaters increased to 20.1% (2393/11884), while the fish scale, conveyor belt and lugs still occupy more than 50% (6124/11884) of the total impacts. The major impact of each surface (maximum peak G) was also evaluated for its relative portion to the total, the fish scale, conveyor belt and lugs (phase 2 and phase 4) increased to 77% (6124/7944) of the total, as shown in the third bar.

The number of impacts generated at different surfaces was plotted in the last bar of Figure 4.4. The BIRD sensor experienced 65% (97/149) of the impacts in phase 2 and phase 4(fish scale, conveyor belt and lug). Sensor's impacts with the beaters accounted for 10.7% (16/149) of the total number of impacts. There were also 16.8% (25/149) of impacts that were not identified by the video, which could possibly be the impacts with bush limbs, the beaters and tunnel.

Evaluation of all four parameters (Bar 1, 2,3 and 4) confirm that the majority of the impacts are located on the fish scale, the conveyor belt and the lug, while the impacts with the beaters and the tunnel only contributed ≤20% of the total impacts, including both number of impacts and the accumulative peak G. There was no impact during phase 3. The fish scale, conveyor belt and lugs are the critical points that lead to bruising of the blueberry. Previous methods, such as providing a padded surface above the conveyor belt area can only address 10% (28/149) of the total impacts, while the majority of impacts on other surfaces still exist. Padding the fish scales can be difficult for those tested harvesters as the space between the overlapped fish scales was too small. Changing the mounting angle of the fishers scales to allow extra room for applying the padding materials is one solution. Reducing the impact at the lugs can be achieved by reducing drop height from the conveyor belt to the lugs and provide cushion pads for the lug surface. The method of using the close-up video and the BIRD sensor only described the mechanical harvest process for the rotary harvester, which is not necessarily true for other types of harvesters. However, this evaluation method can be used for future study of other types of harvesters.

4.4.2 Comparison of three types of harvesters

The rotary, slapper and sway harvesters were compared with each other under the four designed treatments. The number of impacts, along with the maximum peak G, average peak G and the accumulative peak G were compared. Firstly, comparisons were made among the four treatments for each harvester. Secondly, the three harvester’s differences were compared under the four treatments.

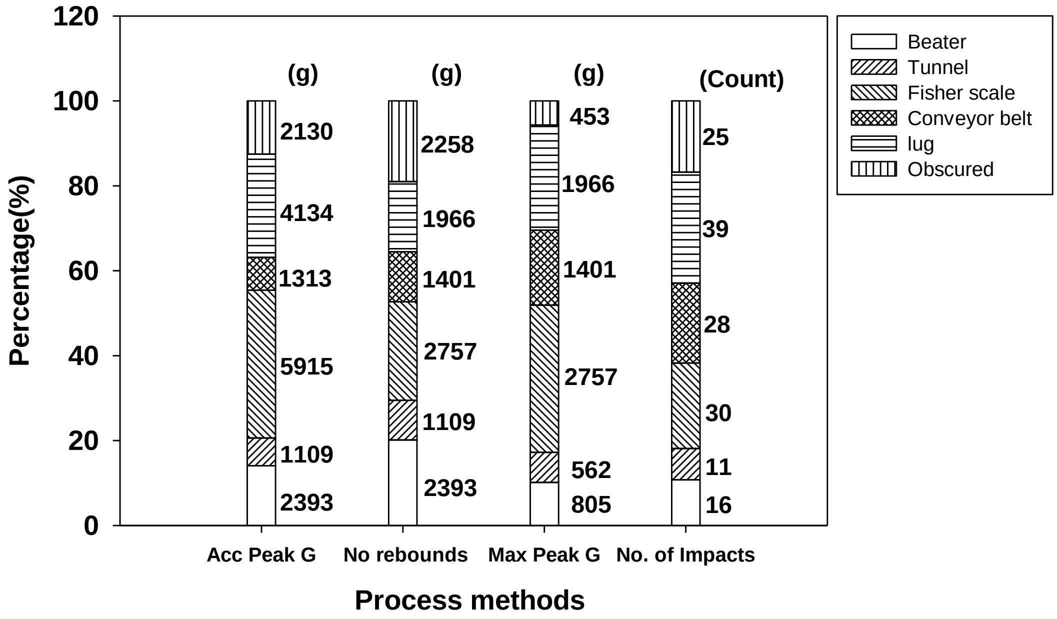


Figure 4.4 Impacts recorded during the rotary's harvest process; Bar 1: Accumulative peak G; Bar 2: Accumulative peak G without rebounds; Bar 3: Sum of maximum peak G; Bar 4: Number of impacts distribution

For the rotary harvester, sensors mounted at the high position generated more impacts than the low mounted treatments based on the comparison at the same speed (H-F 22.75 to L-F 19.7; H-S 22.7 to L-S 16.55). Statistically, the difference is significant. The possible reason is that the high density of rotary fingers extend from the vertical spindles were more likely to hit the falling sensor when the sensor was dropped from a higher position. The two beater speeds did not create significant difference between the high mounted treatments, but create minor difference between the low mounted sensors for the rotary harvester. For the slapper (31~34) and sway harvesters (30~38), no significant difference have been found among all four treatments, which shows that neither mounting height nor beater speed can affect the number of impacts for those two harvesters. Possible explanation of this, as observed in the field, is the slapper and the sway only have one or two columns of beaters extended from the vertical mounted axis, so the sensor may not impact with beaters with significant number of impacts when mounted at the high position.

In conclusion, for the rotary harvester, the two high-mounted treatments have 3~6 more impacts than the low mounted treatments due to the high density of the rotary fingers, while beater speed cannot affect the number of impacts. Both mounting height and beater speed did not create significant difference for the slapper and sway harvesters. However, fast beater speeds do create limb damage for the latter two harvesters, as observed in the field. Also, the rotary harvester generated significantly less number of impacts than the other two machines, indicating a less chance of bruising for the blueberry fruits.

For the difference among three harvesters, the slapper and the sway harvesters (30.5~34.5) generated 10 more impacts on average than the rotary harvester did (19.7~22.8) in the first three treatments (H-F, H-S, L-F), doubled the number of impacts of the rotary harvester (slapper and sway 34~38 to rotary 16.5) in the last treatment. The slapper and sway harvesters have no significant difference in all treatments, yet they all statistically different from the rotary under each treatment.

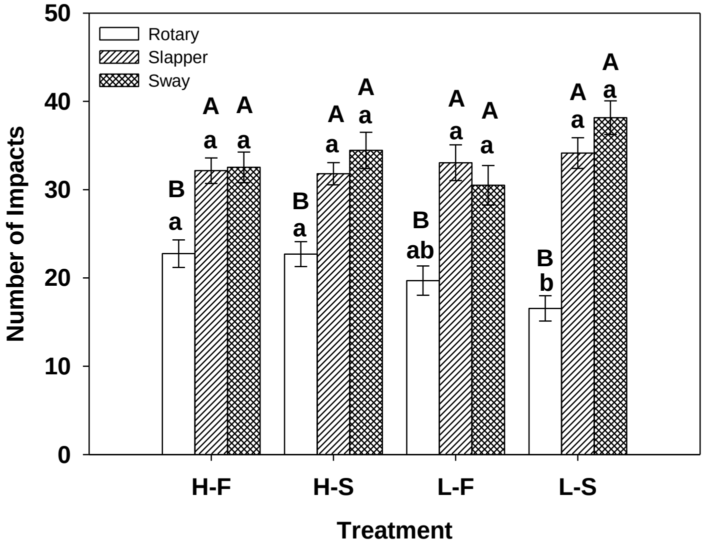


Figure 4.5 Numbers of impacts for rotary, slapper and sway under four treatments; H: High mount, L: Low mount, F: Fast beater speed, S: Slow speed. The same lower case letters indicate there is no significant difference among the four treatments of one harvester. The same upper case letters means there was no significant different among the three harvesters under one treatment.

4.4.2.2 Maximum Peak G analysis

The maximum peak G indicates the severest impact that the sensor experienced during one complete harvest process. The results are provided in Figure 4.6(a).

The rotary harvester had very close mean value of maximum peak G (344~357 g) among all treatments and statistically they were not significantly different. The slapper machine had higher maximum peak G among the high mounted treatments than the low mounted treatments (H-F 508 g to L-F 375 g; H-S 459 g to L-S 372 g). The statistical result also confirms this difference. For different beater speeds, no significant difference exists between the high mounted treatments (H-F 508 g to H-S 459 g), and no statistical difference was observed between the low mounted treatments(L-F 375 g to L-S 372 g). The sway harvester also have higher maximum peak G in the high-mounted treatments than the low mounted ones (H-F 538 g to L-F 468 g; H-S 486 g to L-S 407 g). The fast beater speed also generated higher maximum peak G than the low speed did (H-F 538 g to H-S 486 g; L-F 468 g to L-S 407 g). However, no statistical difference has been found among all four treatments. The sway's L-F treatment shows a very large variation as only 4 replicates were available.

In conclusion, for rotary harvester, neither the sensor's mounting height nor the beater speeds can affect the maximum Peak G, which indicated that the most serve impact during one replicate was not generated with the beaters. On the other hand, if the first contact with the fish scale was the severest impact, the high mounted sensor may be decelerated when impact with the high density of beating fingers, which may mitigate the severity of that impact, so the peak G would not be significantly different. The mounting height can affect slapper and sway's maximum peak G. The beater speed can generate minor difference for the slapper and sway's maximum peak G, but without significant difference statistically. The slapper and sway beaters can create high beater speed, which hit the sensor severely during its falling process. So, the high mounted sensors have more chance of receiving such hits. Fast beater speed also generated higher maximum peak G for the slapper and sway machines due to the increased beating speed. The rotary harvester did not show this phenomenon because the beaters do not have such active spinning speed, so sensors were unlikely to be impacted severely.

For the differences between the three harvesters, in the low mounted treatments, the three harvesters were close to each other in maximum peak G, while for the high mounted sensors, the slapper and sway harvesters have significantly higher maximum peak G than that of the rotary harvester. The reason is the same as explained previously.

4.4.2.3 Average Peak G analysis

The average peak G was calculated as the mean value of all impacts during one replicate. The average peak G indicates the average severity all impacts happened in one replicate.

Both the rotary (83~98 g) and sway (93~112 g) have no statistical difference among the four treatments. So neither mounting high nor beater speed can affect the average peak G for them. It implied that the average severity of impacts has no relation with the beater's speed and the sensor's position for the two harvesters. Although the rotary generated more numbers of impacts for high mounted treatments, the peak G of those impacts didn't affect the overall average peak G. The sway had no significant difference in number of impacts among all treatments, which showed that the higher maximum peak G of high mounted treatments was not significant enough to affect the average peak G of all impacts. For the slapper harvester, the high mounted treatments had significantly higher average peak G than that of the low mounted treatments (H-F 110 g to L-F 92 g; H-S 100 g to L-S 89 g), but beater speed cannot generate different average peak G.

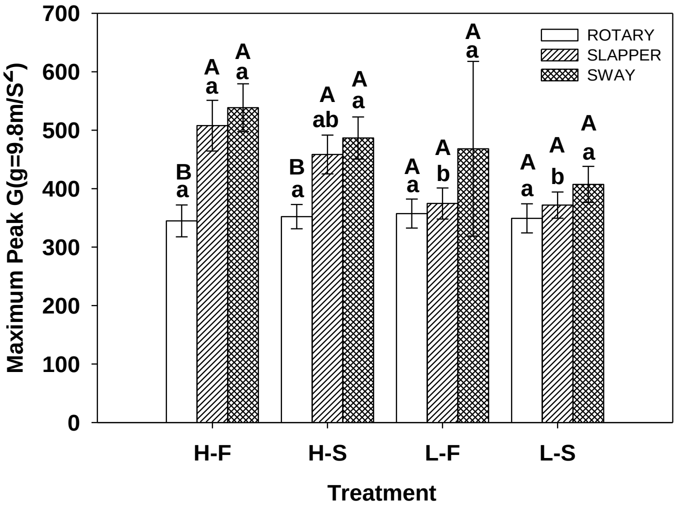
For the differences among three harvesters, the low mounted treatments had very close average peak G for all three harvesters, while for the high mounted ones, the slapper and the sway harvesters had significantly higher average peak G than that of the rotary harvester. Explanation of this phenomenon is that the higher maximum peak G of the high mounted sensors increased the average peak G of the high-mounted treatments in slapper and sway. The average peak G comparisons indicate that the impacts happened in the slapper and sway machines are similar. They differ from the rotary harvester in the high mounted treatments, as the sensor has higher chance severe impacts with the beaters, which increased their average peak G level.

4.4.2.4 Accumulative Peak G analysis

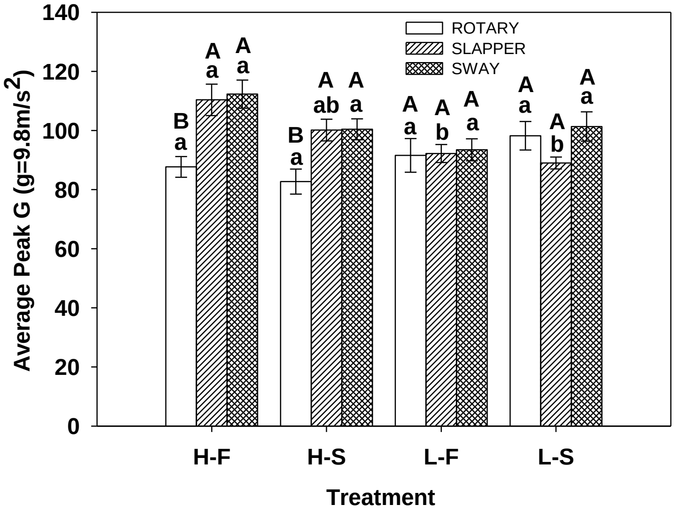
The accumulative peak G, which is the summation of all the peak G values in one replicate, can be an effective way to indicate the impacts' accumulative effect.

None of the three harvesters exhibit significant differences among all four treatments, which indicates that neither mounting height nor beater speed can affect significantly about the harvesters' accumulative peak G. However, for the rotary harvester, the mean value of high mounted sensors can be higher than the low mounted ones (H-F 1948 g to L-F 1684 g; H-S 1803 g to L-S 1584 g). The fast beater speed also generated higher accumulative peak G than that of the slow beater speed (H-F 1948 g to H-S 1803 g; L-F 1685 g to L-S 1584 g). The slapper shows the similar pattern as the rotary. For the slapper harvester, the high mounted treatments have higher accumulative peak G than low mounted treatments (H-F 3484 g to L-F 2988 G; H-S 3165 g to L-S 3034 g). The fast speed created higher accumulative peak G than the slow speed in the high mounted sensors (H-F 3484 to H-S 3165), but very close value at the low mounted treatments (L-F 2988 to L-S 3034). The sway harvester have similar mean value of accumulative peak G with the slapper in H-F, H-S, L-F but have much higher values in the last treatment(L-S 3748).

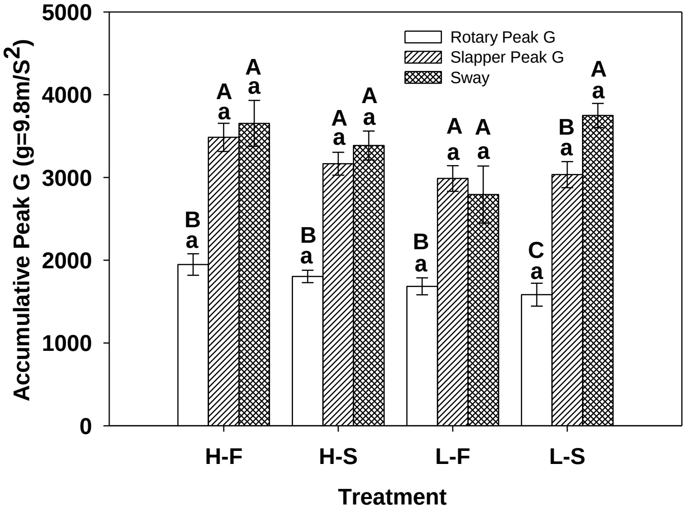
The accumulative peak G created by the the slapper and sway harvesters were about two times of that created by the rotary harvester in all four treatments. For the first three treatments, statistical difference has been found between rotary and the other two harvesters (slapper and sway), but not between the slapper and the sway. All three harvesters were statistically different from each other in the last treatment (L-S). This comparison shows that the rotary was the least violent harvester from accumulative effects of impacts, while slapper and sway have similar performance.



(a)



(b)



(c)

Figure 4.6 Comparison of peak G for rotary, slapper and sway harvesters under four treatments;(a) Maximum peak G, (b) Average peak G (c) Accumulative peak G H: High mount, L: Low mount, F: Fast beater speed, S: Slow speed. The same lower case letters indicate there is no significant difference among the four treatments of one harvester. The same upper case letters means there was no significant different among the three harvesters under one treatment.

4.4.2.5 Overall comparison of three harvesters

The three harvesters were also compared to each other without any treatments differences by averaging all replicates of four treatments. As shown in Figure 4.7, the mean values of the maximum peak G, average peak G and accumulative peak G follows the same trend: Sway> Slapper> Rotary. The statistical result shows that the slapper and sway harvesters had significant difference in accumulative peak G, but not in maximum peak G and average peak G. The rotary has significant difference with the other two harvesters in all three parameters. It has the least maximum peak G, average peak G and accumulative peak G, which indicates the least bruising among all three harvesters.

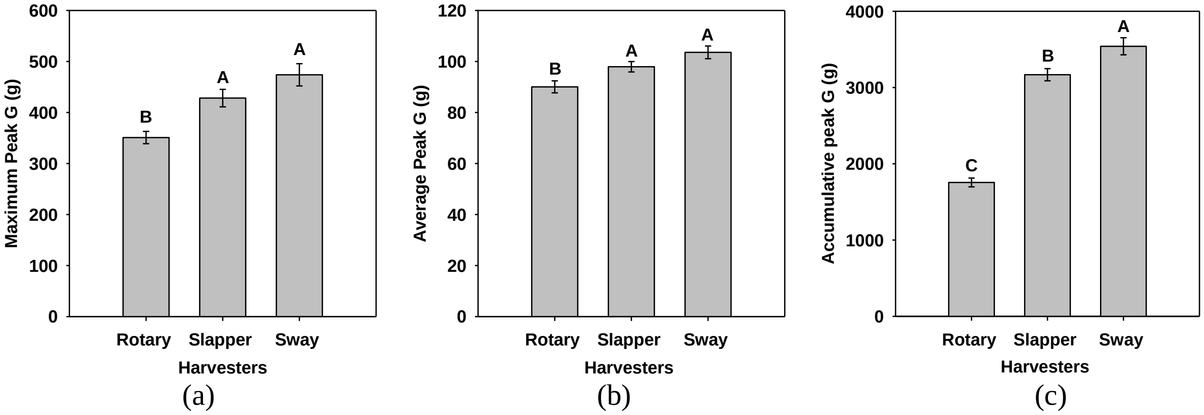


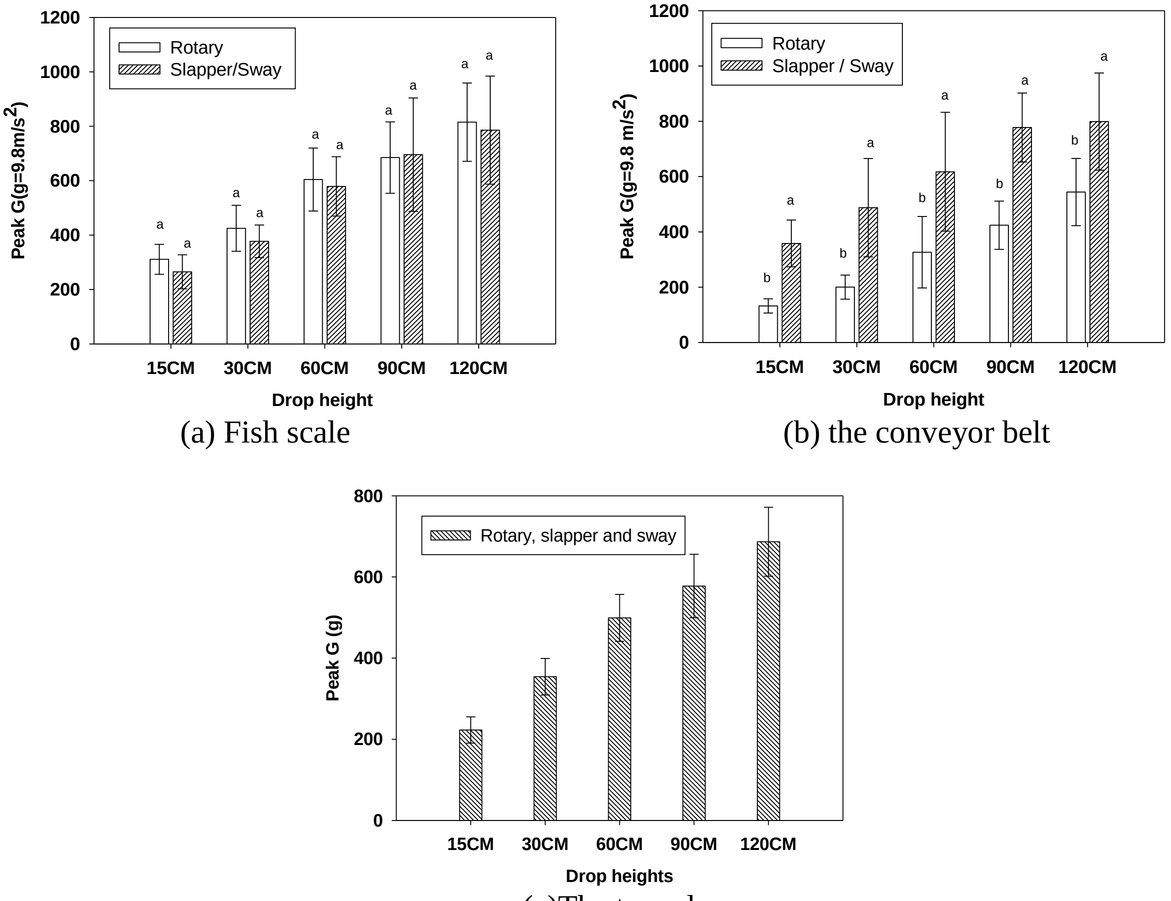
Figure 4.7. Peak G comparison of three harvesters with average of all replicates

4.4.2.7 Harvester surface material

Based on the measured dimensions, the fish scale occupies 80% of the harvester's bottom surface, so it can impact with the majority of the blueberries collected by the harvester. Figure 4.8(a) shows their performance. The error bars indicate the standard deviation of peak G for each dropping height. Peak G of the two types of fish scales had no statistical difference in all five heights. The mean values increased from 265 g to 800 g with the increasing of the drop height from 15 cm to 120 cm.

As shown in Figure 4.8 (b), at each drop height, the peak G shows significant difference between the plastic (rotary harvester) conveyor belt (132 ~544 g) and the steel (slapper and sway harvesters) conveyor belt (358 ~799 g). The steel conveyor belt generated much higher peak G than the plastic conveyor belt at each drop height.

Figure 4.8(c) provides the peak G of the steel tunnel, which represents all three harvesters. The peak G also increases with increment of drop height (223~ 686 g).



(a) Fish scale

(b) the conveyor belt

(c)The tunnel

Figure 4.8 Comparison of rotary, slapper and sway harvesters' surface properties; The same lower case letters indicate that there is no significant difference between the peak G of the two surfaces at the given height

The ranking of all surfaces based on the peak G can be listed as follows:Fish scale (rotary) ≈ Fish scale (slapper/sway)>conveyor belt (slapper/sway)>Tunnel>conveyor belt (rotary). As all tested surfaces were made of rigid hard materials, a higher peak G indicates a severe impact. The Signal-Noise Ratio(SNR) for those tested results was evaluated measure the large variation. The SNR for the fish scale, conveyor belt (rotary), conveyor belt (slapper, sway) and steel were 4.74, 4.32, 4.63 and 7.77, respectively. The overall low SNR was because of the non-uniformity of the sensor housing, the sensor's output can be dependent on the orientation angle of the impacts, especially on those hard surfaces. The reason of higher SNR for the steel surface is a pendulum swing was used to test the vertically mounted steel tunnel, which reduced the manual errors introduced during drop test.

4.4.3 BIRD sensor data correlation with blueberry bruising data

4.4.3.1 Comparison of different cultivars in bruising performance

Susceptibility of blueberries to impact damage can vary among the four different cultivars, which can be categorized into two types: melting or crispy flesh. The melting varieties can be much softer than the crispy flesh types. The scintilla was a cultivar belongs to the melting while all other three cultivars were crispy flesh. Four cultivars of blueberries were dropped to the fish scale surface to evaluate their bruising rates, as shown in Figure 4.9. The "Scintilla" showed the highest bruising rate (17%~78%). The "Farthing" (10%~ 64%) and "Sweet crisp" (10% ~ 68%) showed lower bruising rate than "Scintilla" in both initial bruising rate and two elevated drops. The cultivar 05-258 showed much lower bruising rate than the other three cultivars (2% ~31%). This result confirmed that Scintilla can have much higher susceptibility of bruising to those impacts. The result also differentiated the other three cultivars' damage tolerance under the same drop conditions. The new cultivar 05-528 was confirmed with much higher tolerance of to impact damages, which can be potentially harvested using mechanical harvesters. The initial bruising could be introduced during the picking and handling process. Berries were detached from the clusters using thumbs and fingers and placed into a 1-gal plastic pail, which could potentially be equivalent to a drop height of 0~1 ft.

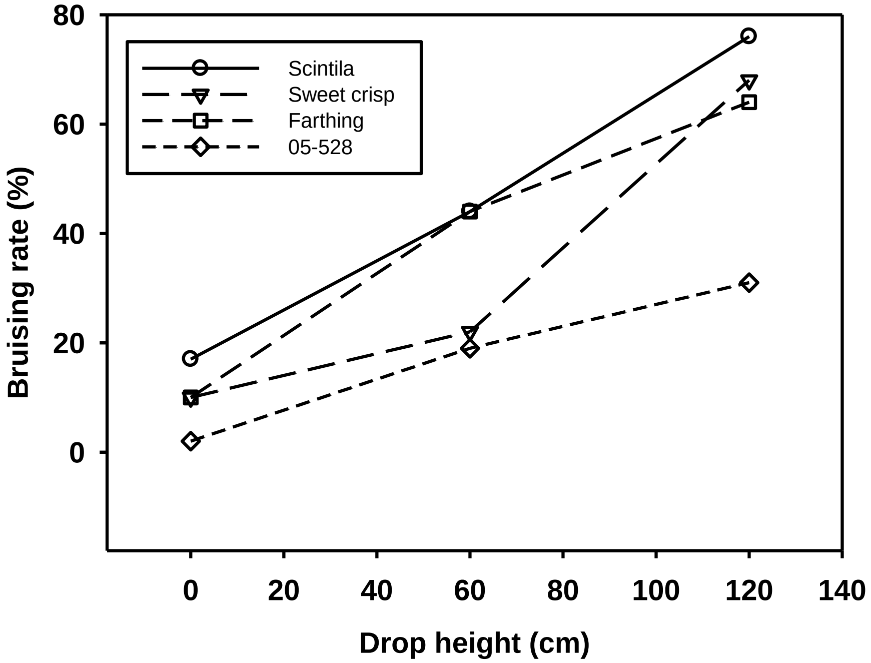


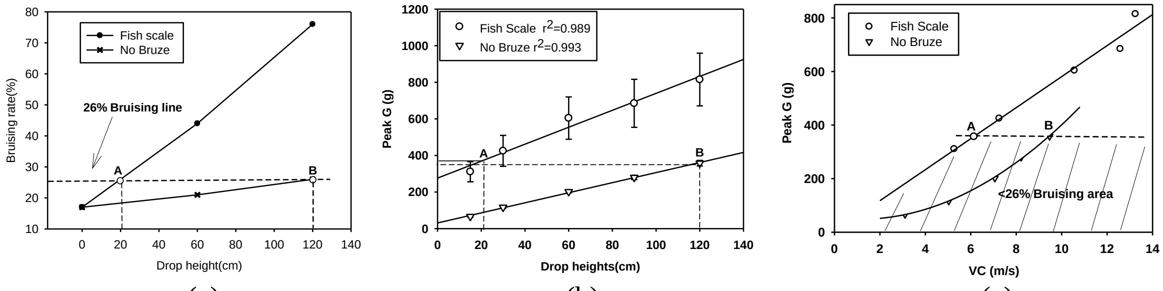
Figure 4.9 Bruising comparisons of four different cultivars of blueberry fruits

4.4.3.2 Correlation between sensor data and the blueberry bruising probability

The BIRD sensor was dropped onto fish scale and “No Bruze” surfaces at five different heights (15 cm, 30 cm, 60 cm, 90 cm, 120 cm). For each surface at each drop height, the BIRD can record the respective Peak G and VC. The regression lines of the Peak G to VC and Peak G to drop height provided the characterization line for both surfaces.

The relationship between the bruising rate of blueberry and the drop height was also achieved based on the drop test of blueberry in three cultivars. Two drop heights were selected (2 ft and 4 ft), plus the control test that provided the initial bruising rate of blueberry, three points were used in total. Straight lines were used to link the three points to provide the relationship between the bruising rate of blueberry and the drop heights.

The correlation between the sensor data and bruising rate of blueberry can be correlated based on the same drop height on each surface. The correlation for “Scintilla” was performed first. As no significant difference has been found among the non-dropped and 2 ft drop berries on the No Bruze surface, the bruising rate (26%) of the maximum drop height (120 cm, “B” in Figure 4.10) was used, as shown in Figure 4.10(a). The bruising rate (26%) of this point is the maximum bruising rate when drop height is less than 120 cm on the No Bruze surface. On the other hand, the 26% bruising rate’s equivalent drop height can be acquired for the fish scale surface, with result of 21.0 cm (“A” in Figure 4.10). The corresponding peak _G and the sensor dropped at the same heights (21 cm to fish scale, 120 cm to No Bruze) can be identified, as it showed in Figure 4.10 (b). Based on the acquired peak G at each surface, the two corresponding points in Figure 4.10(c) can be identified (A, B). By linking point A and point B, the shaded area serves as a area where impacts points can generate bruising rate less than 26%. For area beyond that, the bruising rate can’t be assured. To clearly identify the actual bruising rate line, higher drop heights on the No Bruze surface need to be tested.



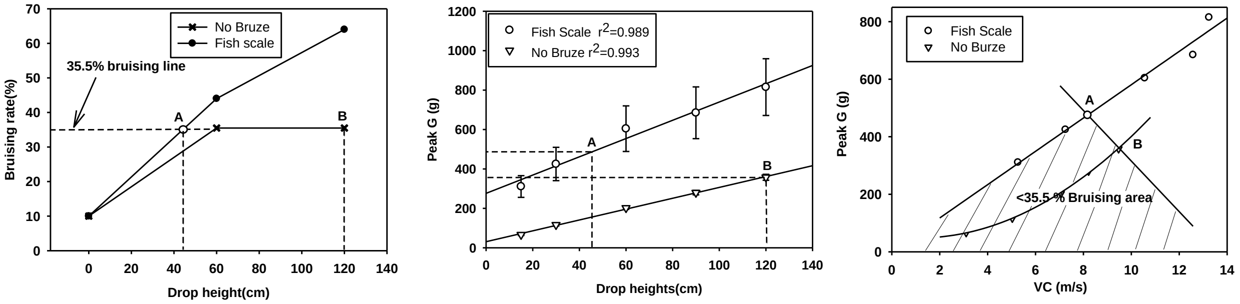
(a)

(b)

(c)

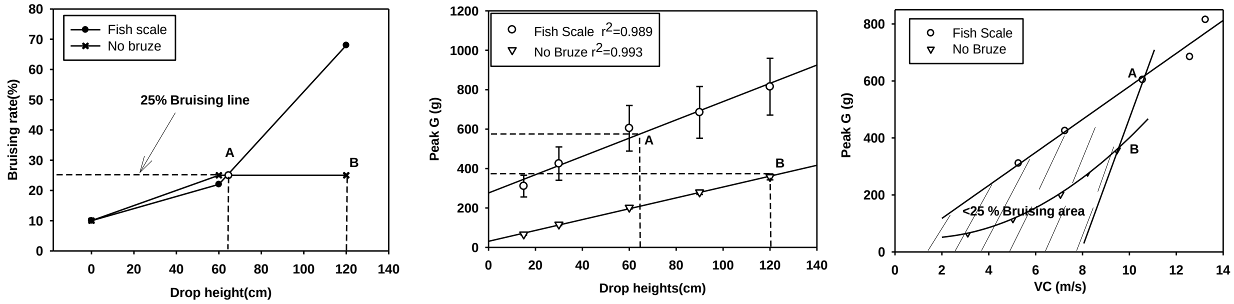
Figure 4.10. Bruising correlation of “Scintilla” blueberry fruits; (a) Bruising probability of the “Scintilla” along with drop heights (b) Identify the peak G based on the drop heights using BIRD Peak G_Drop height plot (c) 26% Bruising threshold line of “Scintilla” based on peak G_VC plot

Bruising correlation was also established for the cultivars “Farthing” and “Sweet crisp”. Those two varieties did not show significant difference of bruising rate between the two drop heights on the padded surface. So, the bruising rate at 2ft drop and 4 fp drops were averaged as the bruising rate for both drop heights (35.5% for Farthing, 25% for Sweet Crisp). The bruising rates at the highest point (120cm) of the No Bruze surface (B) were also used as threshold points for Farthing and Sweep Crisp. The equivalent drop heights can be identified on the hard surface(A), for “35.5%” bruising rate of Farthing and 25% bruising rate of sweet crisp, respectively, as shown in Figure 4.11(a), and Figure 4.12(a). Following the same method as the scintilla, the peak G identified in 4.11(b) and 4.12(b) can be used to pinpoint the points in Figure 4.11(c) and Figure 4.12(c). The shaded area in both Figures indicated a “less bruising” area than the threshold.



(a) (b) (c)

Figure 4.11 Bruising correlation for the “Farthing” blueberry fruits; (a) Bruising probability of the “Farthing” along with drop heights (b) Identify the peak G based on the drop heights using BIRD Peak G_Drop height plot (c) 35.5% Bruising threshold line of “Farthing” based on Peak G VC plot



(a) (b) (c)

Figure 4.12. Bruising correlation of "Sweet crisp" blueberry fruits; (a) Bruising probability of the "Farthing" along with drop heights (b) Identify the peak G based on the drop heights using BIRD Peak G_Drop height plot (c) 25% Bruising threshold line of "Farthing"

With this correlation, the impacts that collected in the field can be plotted on those bruising correlation figures. Impacts points inside the shaded area can create less bruising rate than the threshold line. As the three examples of correlation were not complete, the bruising rate of impacts points locate at the outside of the shaded area can't be predicated. To acquire a complete correlation, higher drop heights are needed to acquire the actual extrapolation of blueberry bruising rate against the drop heights on the padded surface.

4.5 Conclusion

The BIRD impact record sensor was successfully applied in the field to collect impact data during the blueberry mechanical harvest process. The results of this research are summarized below:

1. The blueberry mechanical harvester process can be quantitatively described by the BIRD sensor. The sensor's dynamic experiences were recorded by a close-up video, which were synchronized with the sensor data and used to describe the dynamic impacts numerically. Sensor's experiences inside the mechanical harvest process were classified into four continuous phases and impact data for each phase were analyzed and compared. The fish

scale, conveyor belt and the lug were three critical surfaces that generated the majority of the bruising (≥65%). The impacts with the beaters and the tunnel were less significant (≤20%).

2. The Rotary, slapper and sway harvesters were compared based on the data collected by the BIRD sensor. The Rotary harvester was proven to have the best performance. It had both the least total impacts and the least accumulative peak G. The sway harvester recorded the highest number of impacts among the three harvesters, but statistically, it had close performance with the slapper harvester
3. BIRD sensor data were correlated with blueberry bruising rates for three cultivars: Farthing, Scintilla, and Sweet crisp. Based on this correlation, the BIRD data collected in the field can be used to predicate its bruising rate of actual blueberries.

CHAPTER 5

CONCLUSIONS

This thesis described the BIRD system that was specifically designed for evaluating blueberry's bruising damage in the mechanical harvest process. Both the hardware and the software of this system met our initial design requirements. The system was also successfully applied in the field to evaluate different types of mechanical harvesters.

To summarize this study, the following conclusions can be made:

1. A miniature impact recording sensor was designed and calibrated. The circuit board of the sensor has a diameter of 19.4 mm. It has three axes of MEMS accelerometers, one MCU, and one 1-Mbits non-volatile memory. After casting the circuit board into a one inch rubber sphere, the sensor is a standalone unit that can record three axes of acceleration data with accuracy of -0.53~0.33% over the range of ±500 g and precision error of 0.63%. The sensor is powered by a rechargeable battery that lasts for three hours. An interface box was also designed to connect the sensor with the computer.
2. A complete data acquisition, processing and analyzing software platform was designed and tested for the BIRD system. The data sampling software that operates on the BIRD sensor has a maximum sampling rate of 3.0 KHz. The sensor records acceleration data as single impacts. The recorded impacts were verified by comparing with the same impact curve that recorded by a high performance NI-DAQ data logger. The sampling software can retain the impacts' original shape with low distortion. The velocity change of the

recorded impact curve has an error less than 5% at the three tested frequencies. The interface program can establish the communication between the sensor and the computer. It also updates the LCD to display the operation status. The i-BIRD computer software was designed under LabVIEW graphical programming environment. It can configure the sensor with a sampling threshold and sampling frequency, synchronize the sensor's clock with the computer. The i-BIRD can be used to download and process data, and graphically display the results.

3. The blueberry bruising damage and three types of mechanical harvesters were evaluated by the BIRD sensor. Firstly, a rotary harvester's harvest process was quantitatively described. In this test, the video record registered the actual location of each impact, which was also recorded by the BIRD sensor. This provided a quantitative description about the sensor's impacts on each surface. The result revealed that the fish scale, conveyor belt and the lug surface are three critical surfaces that generated the majority of the impacts (>65 %). However, the impacts with the beaters were less significant (≤20%). The sensor's response was also correlated with the bruising rate of blueberries for three cultivars: "Farthing", "Scintilla" and "Sweet crisp". Bruising probability of different cultivars was also compared. Finally, the rotary, slapper and sway harvesters' overall performance were compared. The rotary harvester was proven to cause the least bruising damage among the three harvesters. It generated the least number of impacts, lowest accumulative peak G and lowest maximum peak G. The sway and slapper harvester have similar performance statistically, but the sway has a slightly higher number of impacts than the slapper.

The field application result shows that the BIRD system is a useful tool to study the bruising of blueberries in the mechanical harvest process. However, there are several issues that should be addressed in the future studies:

1. The design of sensor can be further improved in the following three aspects:
    a. The measured peak G values showed a relative strong dependence on the orientation angle of impacts, especially on the hard surface. This is due to the non-uniform thickness of the sensor’s rubber housing at different orientations. A potential solution is to increase the thickness of the rubber along the equator region of the sensor.
    b. The sensor’s robustness in the field should be improved. Due to harsh operations in the field, the sensor was occasionally damaged; the power supply of certain sensor was also unstable in several cases.
    c. It would be ideal if the size of the BIRD sensor can be further reduced to 15 mm in diameter, with the reducing size of electronic components, a more efficient battery and a more compact circuit design.
2. The correlation between the sensor data and bruising probability of berries can be further enhanced. Firstly, singe impact’s correlation can he improved. More drop heights are needed to acquire accurate extrapolation between bruising rates of blueberry and the drop height. Secondly, the damage potential of the accumulative effect of multiple impacts was not verified. Onsite correlation between the sensor’s data and the bruising rate of blueberry can be useful information to evaluate the accumulative effects of all impacts. Lastly, the hand harvest process can be tested using the BIRD sensor as a baseline to

compare with the mechanical harvesters. It can also be used to analyze the initial bruising rate of blueberries that are hand harvested.

REFERENCES

Bajema, R.W.; Hyde, G.M. (1995). "Packing Line Bruise Evaluation For 'Wallla Walla' Summer Sweet Onions." Transactions of ASAE **38**(4): 1167-1171.

Bollen, A. F. (2006). "Technological innovations in sensors for assessment of postharvest mechanical handling system " International Journal of Postharvest Technology and innovation **1**(1): 16-31.

Bowden, E.P. and Tabor, D. (1954). "The Friction and Lubrication of Solids." Oxford University Press.

Brown, G. K., D.E.Marshall, Tennes, B. R. and Booster, D.E. (1983). "Status of harvest mechanization of horticultural crops." American Society of Agricultural Engineers **83**(3).

Canneyt, T. Van, Tijskens, E., Ramon, H., Verschoore, R. and Sonck, B. (2003). "Characterization of a Potato-shaped Instrumented Device." Biosystems Engineering **86**(3): 275-285.

Chiabrando, V., Giacalone, G. and Rolle, L. (2009). "Mechanical Behavior and Quality Traits of Highbush Blueberry during Postharvest Storage." Journal of the Science of Food and Agriculture **89**: 989-992.

Desmet, M., Van linden, V., Hertog, M. L. A. T. M., Verlinden, B. E., De Baerdemaeker, J. and Nicola, B. M. (2004). "Instrumented sphere prediction of tomato stem-puncture injury." Postharvest Biology and Technology **34**(1): 81-92.

G.K. Brown, N.L. Schulte, E.J. Timm, R.M. Beaudry, D.L. Peterson, J.F. Hancock, F. Takeda (1996). "Estimates Of Mechanization Effects On Fresh Blueberry Quality." Applied Engineering in Agriculture **12**(1): 21-26.

Geyer, M. O., Praeger, U., König, C., Graf, A., Truppel, I., Schlüter, O. and Herold, B. (2009). Measuring Behavior of an Acceleration Measuring Unit Implanted in Potatoes. **52:** 1267-1274.

Goldsmith, W. (1960). "Impact, The Theory and Physical Behavior of Colliding Bodies." Edward Arnold Publishers.

Herold, B., Oberbarnscheidt, B. and Geyer, M. (1998). "Mechanical Load and its Effect on Bulb Onions due to Harvest and Post-harvest Handling." Journal of Agricultural Engineering Research **71**(4): 373-383.

Herold, B., Truppel, I., Jacobs, A. and Geyer, M. (2005). "Stosdetektor zum implantieren in empflindliche fruchte=Impact detector for implantation into perishable fruit." landtechnik **60**(4): 208-209.

Herold, B., Truppel, I., Siering, G. and Geyer, M. (1996). "A Pressure Measuring Sphere for monitoring handling of fruit and vegetables." Computers and Electronics in Agriculture **15**(1): 73-88.

Klug, B.A., Tennes, B.R., Zapp, H.R., Siyami, S. and Clemens, J.R. (1987). "Software for a Miniature Impact Data Acquisition Device " Transactions of ASAE **30**(6): 1818-1821.

Lin, X. and Brusewitz, G.H. (1994). "Peach Bruise Thresholds Using The Instrumented Sphere." Applied Engineering in Agriculture, American society of Agricultural Engineers **10**(4): 509-513.

Mainland, C.M. (1993). "Blueberry Production Strategies." Acta Hort.(ISHS) **346**: 111-116.

Mathew, R. and Hyde, G.M. (1997). "Potato impact damage thresholds." Transactions of the ASAE **40**(3): 705-709.

Meththananda, Iranthi M., Parker, Sandra, Patel, Mangala P. and Braden, Michael (2009). "The relationship between Shore hardness of elastomeric dental materials and Young's modulus." Dental Materials **25**(8): 956-959.

Microchip. "PIC18LF2520." http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010277.

Mohsenin, N.N. (1980). Physical Properties of Plant and Animal Materials, Dept. of Agricultural Engineering, Pennsylvania State University in [University Park] .

Prussia, S.E., Tetteh, M.K., Verma, B.P. and NeSmith, D.S. (2006). "Apparent Modulus of Blueberry Elasticity From FirmTech2 Firmness Measurements of Blueberries." Transactions of ASABE **49**(1): 113-121.

Schulte, N.L., Brown, G. K. and Timm, E.J. (1992). "Apple Impact Damage Thresholds." Applied Engineering in Agriculture **8**(1): 55-60.

Schulte, N.L., Timm, E.J. and Brown, G. K. (1994). "'Redhaven' Peach Impact Damage Thresholds." HortScience **29**(9): 1052-1055.

Simami, S. , Tennes, B.R., Zapp, H.R., Brown, G. K., Klug, B.A. and Clemens, J.R. (1987). Microcontroller-Based Data Acquisition System for Impact Measurement. Transactions of ASAE. **30:** 1822-1826.

Strik, B. (2006). "Blueberry Production and Research Trends in North America." Acta Hort.(ISHS) **715**: 173-184.

Strik, B. and Yarborough, D. (2005). "Blueberry production trends in North America, 1922 to 2003, and predictions for growth." HortTechnology **15**(391-398).

Takeda, F, Krewer, G, Andrews, E.L. and Mullinix, B (2008). "Assessment of the V45 Blueberry Harvester on Rabbiteye Blueberry and Southern Highbush Blueberry Pruned to V-Shaped Canopy " HortTechnology **18**(1): 130-138.

Wheeler, Anthony J. and Ganji, Ahmad R. (1996). "Introduction to Engineering Experimentation." 18-24.

Zapp, H. R., Ehlert, S.H., Brown, G. K., Armstrong, P. R. and Sober, S.S. (1990). "Advanced Instrumented Sphere(IS) For Impact Measurements." Transactions of ASAE **33**(3): 955-960.

APPENDICES

Appendix A

Source code of BIRD

1. File name: BIRD_sensor

```
'****************************************************************
'*  Name    : BIRD_Sensor.BAS                                   *
'*  Author  : [PENGCHENG YU]                                     *
'*  Notice  : Copyright (c) 2010                                 *
'*          : All Rights Reserved                                *
'*  Date    : 10/14/2010                                         *
'*  Version : 2.8                                                *
'****************************************************************
'
'###########################################################################
####'
'INITIALIZATION SETUP FOR SYSTEM VAIRABLES,REGISTERS, SYMBOLS
'###########################################################################
####'

'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
CLEAR
define OSC 20      'Define the Cystal Speed, 20MHz
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'Defination of Flags and Threshold Variables'
Fre_F           var bit       ' Frequency Set Flag
Thre_F          var bit       ' Threshold value set Flag
Main_F          var bit       ' Flag for the data reading and write cycle
Stop_F          var bit       ' Flag for stop data reading and write cycle
Stop_F2         var bit       ' Flat for the reading SPI memory and writing I2C memory cycle.
LT_F            var bit       ' Leader or trailer writing flag
RTC_F           var bit       ' Flag for synchronization
Thre_V          var WORD      ' Data Record Threshold value
Pau_T           var word      ' pause time for frquency adjustment
Pow_DT          var word      ' Power down time(Seconds)
Com_F           var byte      ' The compare flag
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
L_UD              var byte[48]   ' Leader updated array, two leaders, each leader has 11bytes
L_P              var byte       ' Leader pointer
DD               var byte[240]  ' Array for Impact data storage(Values above threshold)
D_P              var byte       ' Data pointer
T_D              var byte[96]   ' Trailer data storage array
T_P              var byte       ' Trailer Pointer
T_C              VAR BYTE       ' Trailer count
L_C              var byte       ' Leader count
I_I             VAR word       ' Impact Index
'-------------------------------------------------------------------------------
L_UD=0;
L_P=0;
DD=0;
D_P=0;
T_D=0;
T_P=0;
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
readcnt            VAR BYTE      ' These three variables are used for I2C communication
datain             VAR BYTE      ' I2C reading data
WrData             VAR BYTE      ' I2C data reading status
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'Control word received from interface board
' The I2C Master-Slave communication contains the setup information for the
' sensor.It contains ten bytes.With last byte to determine the type of parameter.
' With certain number of other bytes(LSB)to show the actual control data. See command
' list for detail list.
CON_P              VAR byte      ' Determines what type of control information(The first byte
of CON_BUFF)
'Command List for PC-Interface-Sensor communications
'   9  8  7  6  5  4  3  2  1  0    Digits
'   X  X  X  X  X  X  X  X  X  0    Download FRAM Data(Sensor board) to Interface board
memory
'   X  X  X  X  X  X  a  b  c  1    Setup threshold acceleration=a*100+b*10+c
'   X  X  X  X  X  X  X  X  X  2    Setup the frequency,this is temporarily not implanted, but
need to be setup to remove the flag
'   X  X  X  X  X  X  X  X  X  3    Download interface memory data and serially sent to PC
'   X  X  X  X  X  X  X  X  X  4    Activate the sensor and start sampling and recording data
'   X  X  X  X  X  X  X  X  X  5    Erase interface memory data
'   X  X  X  X  X  X  X  X  X  6    Deactivate the three accelerometer's power supply
'   H  H  M  M  S  S  m  m  m  8    Two digits for hour, two digits for Minute, Two digits for
Second, and three digits for Milisecond
'   X  X  X  X  X  X  X  X  X  9    Erase Sensor Memory !!! Be caution to use this command
during field test, all data will be erased in the sensor
```

```
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
DTPIN            VAR TRISC.3    'I2C slave transfer data pin
CTPIN            VAR TRISC.4    'I2C slave transfer clock pin
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'define control register bits'
SSPIF           var PIR1.3     'SSP(I2C) INTERRUPT FLAG
SSPIE           VAR PIE1.3     'SSP(I2C) INTERRUPT ENABLE FLAG
BF              VAR SSPSTAT.0  'SSP(I2C) BUFFER FULL
R_W             VAR SSPSTAT.2  'SSP(I2C) R/W
D_A             VAR SSPSTAT.5  'SSP(I2C) DATA/ADDRESS
CKP             VAR SSPCON1.4  'CKP: Clock Polarity Select bit
'1 = Idle state for clock is a high level
'0 = Idle state for clock is a low level
SSPEN            VAR SSPCON1.5  'SSP (I2C) Enable
SSPOV            VAR SSPCON1.6  'SSP (I2C) Receive Overflow Indicator
WCOL             VAR SSPCON1.7  'SSP (I2C) Write Collision Detect
STAT_BF            VAR SSPSTAT.0  'SSP (I2C) Buffer Full
STAT_RW             VAR SSPSTAT.2  'SSP (I2C) Read/Write
STAT_DA            VAR SSPSTAT.5  'SSP (I2C) Data/Address
CKE             VAR SSPSTAT.6  'SSP (I2C) Data/Address
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'Define SPI control Registers
FM_CS            VAR PORTA.5    'SPI FRAM CHIP CS PIN
FM_CS_TRIS         VAR TRISA.5    'SPI FRAM CHIP CS PIN DIRECTION
SCK             VAR PORTC.3    'SPI CLOCK
SCK_TRIS          VAR TRISC.3    'SPI CLOCK PIN DIRECTION CONTROL
SDI             VAR PORTC.4    'SPI DATA IN
SDI_TRIS          VAR TRISC.4    'SPI DATA IN PIN DIRECTION
SDO             VAR PORTC.5    'SPI DATA OUT PIN
SDO_TRIS          VAR TRISC.5    'SPI DATA OUT PIN
CONT_WREN         CON %00000110  'F-RAM WRITE ENBALE OP-CODE
CONT_RDSR         CON %00000101  'F-RAM READ STATUS REIGSTER OP-CODE
CONT_WRSR         CON %00000001  'F-RAM WRITE STATUS REGISTER OP-CODE
CONT_READ         CON %00000011  'F-RAM READ MOMEORY OP-CODE
CONT_WRITE        CON %00000010  'F-RAM WRITE MEMORY OP-CODE

TX_BUFF             var byte      'ENDING DATA ARRAY
RX_BUFF             var byte      'RECEIVING DATA ARRAY
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

'Rx Buffer defintion
RxBufferLEN         CON 10        'I2C Receive buffer length
```

```
RxBuffer                  VAR BYTE[Rxbufferlen]
'I2C Receive buffer array
RxBufferIndex          VAR BYTE       'I2C Buffer Reading Index
RxBufferInv             var byte[rxbufferlen]
'I2C Received data invert(this variable is created in invert the digits)
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'Tx Buffer defintion for I2C Master-Slave communication
TxBufferLEN         CON 1          'For salve transfer to master
TxBuffer               VAR BYTE[txbufferlen]'
TxBufferIndex        VAR BYTE       'Buffer data index
I2Caddress             CON $3         'Make address = 3(Sensor prototype's address)

gosub I2C_SLAVE_INIT               'Get the Sensor ready for I2C-Master Slave
communication
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
DPIN                      VAR PORTC.4    'I2C date PIN
CPIN                      VAR PORTC.3    'I2C Clock PIN
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'Time interrup definitions
Ticks              VAR WORD       ' Tick count (61 ticks = 1 sec)
Hour               VAR byte       ' Hour variable
Minute              VAR byte       ' Minute variable
Second             VAR byte       ' Second variable
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
BUFF3              VAR BYTE[64]   ' BUFFER FOR THE READING PROCESS
I                var word       ' Index for Array Operations
Pointer            var word
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
ADD                VAR WORD       ' Address for SPI-Memory address operations
ADDR               VAR WORD       ' Address for Array operations
CON_B              VAR BYTE       ' Writing control byte
BLOCK0             CON %10101000  ' Accessing EEPROM's first block
BLOCK1             CON %10101010  ' Accessing EEPROM's Second Blcok
BLOCK              VAR BYTE       ' Accessing FRAM's address(MSB)  BLOCK=%00000000
CON_B=BLOCK0
ADD=0                         ' Address, pointers intialize
BLOCK=%00000000
'&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
'Variables for sensor data
X                var word
```

```
Y                  var word
Z                  var word
X_D                  VAR WORD
Y_D                  VAR WORD
Z_D                  VAR WORD
X_F                  VAR BIT    'Flags for determine whether data is above threshold vaule
Y_F                  VAR BIT
Z_F                  VAR BIT
'-----------------------Sensor data registers intialization---------------------
X=0
Y=0
Z=0
X_D=0
Y_D=0
Z_D=0
X_F=0
Y_F=0
Z_F=0
'-------------------------------------------------------------------------------

'-------------------------------------------------------------------------------
Thre_V=0               'Default setting of Threshold value
fre_f=0
main_f=0
stop_f=0
stop_f2=0
LT_F=0
'-------------------------------------------------------------------------------


'-------------------------------------------------------------------------------

'-------------------------------------------------------------------------------


'-----------------------A/D setups ------------------------------------------
define        ADC_BITS       10      ' Set number of bits in result
'Define       ADC_CLOCK              3       ' Set clock source (3=rc)
Define        ADC_SAMPLEUS   20          ' Set sampling time in uS
TRISA  = %00001111      ' Set PORTA to all input
ADCON1 = %00001100      ' Set PORTA analog and right justify result THE RIGHT
JUSTIFY DOES NOT WORK PROPERLY
ADCON2 = %10000000   ' SET UP THE CONVERSION CLOCK AND RIGHT JUSTIFED

'ADCON0.7=0;
'ADCON0.6=0;
'ADCON0.0=1;
```

```
'----------------------------Timer Interrupt Handinglings----------------------

' Interrup handling bits
GIE    var INTCON.7
PEIE   var INTCON.6
TMR2IE VAR PIE1.1
TMR2ON VAR T2CON.2
TMR2IF VAR PIR1.1
'T2CON = %01001101 ' Set prescaler = 4; postscale=10 for 4MHz
'T2CON =%01111110  '  For 10MHz PRESCALER=4, POSTSCALE=16
T2CON=%01001100  ' Set the prescaler to 1, and postscale to 10 for 10MHz
'T2CON=%00100101  ' sET THE PRESCALER TO 4, AND POSTSCALE TO 5 FOR
20MHz
T2CON=%00100100 ' Set the prescaler to 1, postscale to 5 for 20Mhz
'T2CON=%00100101
'OPTION_REG = $04 ' Set prescaler = 64
ON INTERRUPT GOTO ISR ' ISR routine
'PR2=249 ' Set PR2 Register for 4MHz
'PR2=156 ' For 10MHz setups
' PR2=224' FOR 1 MILISECOND INTERRUP
PR2=250' adjusted for 10Mhz Crystal
GIE=1
PEIE=1
TMR2IE=1'Enable the Timer2 interrupt.
TMR2ON=1

'********************************  Main
*****************************************
'***************************************************************************
*****

START:

IF SSPIF = 1 Then              ' Check I2C interrup
GoSub i2cslave
EndIF
IF WRDATA=0 THEN GOTO START    ' Not all bytes are received
for i=0 to 9                   ' Reverse the 10bytes of data
RxBufferInv[i]=RxBuffer[9-i]
NEXT I

CON_P=RXBUFFERinv[0]           ' Update the recieved control byte,update CON_P and
determine the control data's type
SSPOV = 0    'clear transfer flags
WCOL = 0     'clear transfer flags
```

```
'~~~~~~~~~~~~~~~~~~~~~~~~~~ Command execution
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Select Case CON_P
Case  1                 ' Updating Threshold Value setup
HIGH PORTB.1
MAIN_F=0        ' Main_F has lower priority
'High LED1
THRE_V=RxBUFFERinv[2]*10+RxBUFFERinv[1] ' Update Threshold value
THRE_F=1        ' Remove Flag
'PAUSE 200
LOW PORTB.1

Case  2                 ' Update Frequency value
HIGH PORTB.1
MAIN_F=0        ' Set Main_F's lower Priority
'HIGH LED2
PAU_T=RxBUFFERinv[4]*10000+RxBUFFERinv[3]*1000+RxBUFFERinv[2]*10+RxBU
FFERinv[1]  ' Update control data to pause intervals
FRE_F=1         ' Remove Flag
'PAUSE 200
LOW PORTB.1

Case 0
HIGH PORTB.1
'PAUSE 1000
LOW PORTB.1
GOSUB SPI_INIT;
gosub SPI_TO_I2C;
GOSUB I2C_SLAVE_INIT;
'PAUSE 200
LOW PORTB.1

case 4
HIGH PORTB.1
main_f=1        ' Remove Main Flag to get started
'PAUSE 200
'LOW PORTB.1
'GOSUB SPI_INIT
case 9
HIGH PORTB.1
GOSUB SPI_INIT
gosub SPI_ERASE;
GOSUB I2C_SLAVE_INIT
'PAUSE 200
```

```
LOW PORTB.1
case 6
if rxbufferinv[1]=1 then high PORTB.5 'Analog Switch on
if rxbufferinv[1]=0 then low  PORTB.5 'Analog Switch Off
IF rxbufferinv[2]=1 then
for I=0 to 65535
LOW PORTB.5
SLEEP 65535
SLEEP 65535
next I
ENDIF

caSE 8 ' Synchronization
Main_f=0
Hour=rxBufferinv[9]*10+rxbufferinv[8]    'Update the hour
Minute=rxbufferinv[7]*10+rxbufferinv[6]  'Update the Minute
Second=rxbufferinv[5]*10+rxbufferinv[4]  'Update the Second
Ticks=rxbufferinv[3]*10+rxbufferinv[2]   'Update the Milisecond
rtc_f=1                                  'Remove Flag for Synchronization
End Select
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~

WrData=0              ' Remove I2C data transfer flag for new cycles

if fre_f && thre_f && main_f && Rtc_f then  'All parameters have been setup
gosub SPI_INIT;
gosub Sen_Re_Write                        'Goto Sub program for writing the memory
'low portb.1
FRE_F=0
THRE_F=0
rtc_f=0
MAIN_F=0
endif

goto START
'**************************************************************************
*****
'**************************************************************************
*****

'----------------------------Sensor Data reading cycle------------------------

Sen_Re_Write:
ADD=0;        ' Clear the Flags and address, ADD: memory address
D_P=0;   Impact Data array pointer
```

```
T_P=0;   Trailer Array pointer
I_I=0;   Impact index
LT_F=0;  Leader or trailer array index
L_C=0;

while STOP_F=0

adcin 0,X         ' ADC converison for channel X
adcin 1,Y         ' ADC conversion for channel Y
adcin 2,Z         ' ADC conversion for channle Z

X_D=ABS(X>>2-127)     ' Get the difference between their zero offset
Y_D=ABS(Y>>2-126)
Z_D=ABS(Z>>2-126)


if sqr(X_D*X_D+Y_D*Y_D+Z_D*Z_D)>=Thre_V   then ' One count equal 5.86G for
Thre_V
'IF X_D>=THRE_V||Y_D>=THRE_V||Z_D>=Thre_V then ' Compare to threshold values if
it is higher than threshold
Com_F=1                                   ' Then Com_F=1, orelse Com_F=0
else
Com_f=0
endif

if LT_F=0 then                     ' Leader updates, L_UD is the array, there are 22bytes(Two
datasets inside)
if L_C<=2||Com_F=0 then;                  ' Judge whether the leader array is full or not
com_f=0;
for I=16 to 28
L_uD[I-16]=L_uD[I]
next I
L_uD[16]=X.byte1
L_uD[17]=X.byte0
L_uD[18]=Y.byte1
L_uD[19]=Y.byte0
L_uD[20]=Z.byte1
L_uD[21]=Z.byte0
L_uD[22]=Hour
L_uD[23]=Minute
L_uD[24]=Second
L_uD[25]=Ticks.byte1
L_uD[26]=Ticks.byte0
L_uD[27]=i_i.byte1
L_uD[28]=I_I.byte0                   ' I_I is Impact Index
L_C=L_C+1;
```

```
'for Pointer=1 to Pau_t
'Pointer=Pointer
'next Pointer
endif
endif


IF Com_F=1&&D_P<240 THEn                    ' Incase data is higher than threshold and the
impact data array is not full
LT_F=1                              ' Clear flag for the trailer
if T_P!=0 then ;                    ' If there are data inside the trailer array then, copy them to
Impact array
'T_C=0;                             ' Clear the Trailer's count if there are impact data recorded
into trailer array, and leader does not reach size of six.
for I=1 to T_P
DD[D_P+I-1]=T_D[I-1];
next I;
D_P=D_P+T_p                             ' updata Impact data pointer
T_D=0;                              ' Trailer data array clear
T_P=0;                              ' Trailer data array pointer clear
endif

IF D_P<240 THEN
DD[D_P]=X.byte1                         'Update the data file
D_P=D_P+1;
DD[D_P]=X.byte0
D_P=D_P+1;
DD[D_P]=Y.byte1
D_P=D_P+1;
DD[D_P]=Y.byte0
D_P=D_P+1;
DD[D_P]=Z.byte1
D_P=D_P+1;
DD[D_P]=Z.byte0
D_P=D_P+1;
DD[D_P]=Hour
D_P=D_P+1;
DD[D_P]=Minute
D_P=D_P+1;
DD[D_P]=Second
D_P=D_P+1;
DD[D_P]=Ticks.byte1
D_P=D_P+1;
DD[D_P]=Ticks.byte0
D_P=D_P+1;
DD[D_P]=I_I.byte1;
```

```
D_P=D_P+1;
dd[D_P]=I_I.byte0;
D_P=D_P+4;
ENDIF
for Pointer=1 to Pau_t
Pointer=Pointer;
next Pointer
endif


IF (LT_F=1&&Com_F=0)||D_P>=240 THEN
' incase the values is less than threshold and it is the trailer or the impact data arry is full
T_D[T_P]=X.byte1                        'Update the data file
T_P=T_P+1;
T_D[T_P]=X.byte0
T_P=T_P+1;
T_D[T_P]=Y.byte1
T_P=T_P+1;
T_D[T_P]=Y.byte0
T_P=T_P+1;
T_D[T_P]=Z.byte1
T_P=T_P+1;
T_D[T_P]=Z.byte0
T_P=T_P+1;
T_D[T_P]=Hour
T_P=T_P+1;
T_D[T_P]=Minute
T_P=T_P+1;
T_D[T_P]=Second
T_P=T_P+1;
T_D[T_P]=Ticks.byte1
T_P=T_P+1;
T_D[T_P]=Ticks.byte0
T_P=T_P+1;
T_D[T_P]=I_I.byte1;
T_P=T_P+1;
T_D[T_P]=I_I.byte0;
T_P=T_P+4;
for Pointer=1 to Pau_t
Pointer=Pointer ;
Pointer=Pointer;
next Pointer
'------------------Data writing to SPI RAM------------------------------------
if T_P=96 then                     ' if the trailer is full, then the impact is complete
FM_CS=0                            ' Start to writing SPI_FRAM
TX_BUFF=cont_WREN      '1
```

```
gosub WriteSPI        '42
FM_CS=1               '1

FM_CS=0               '1
TX_BUFF=cont_write    '1
gosub WriteSPI        '42*14

'-----send three bytes of address-------------
TX_BUFF=BLOCK;
gosub WriteSPI

TX_BUFF=ADD.byte1
gosub WriteSPI

TX_BUFF=ADD.byte0
gosub WriteSPI
'  Write Data bytes

for I=0 to 31   'Write the leaders
TX_BUFF=L_uD[I]
GOSUB WriteSPI
next I

For I=0 to D_P-1
TX_BUFF=DD[I];
gosub WriteSPI;
next I

For I=0 to 95
TX_BUFF=T_D[I];
gosub WriteSPI;
next I
FM_CS=1;

ADD=ADD+128+D_P;    updata the memory address

I_I=I_I+1;        update the index
LT_F=0;
'L_UD=0;
DD=0;
'T_D=0;
T_P=0;
D_P=0;
L_C=0; 'LEADER COUNTER CLEAR TO ZERO
endif
'--------------------------------------------------------------------------
```

```
endif

if ADD>=63500&&BLOCK=%00000000 then
BLOCK=%00000001
ADD=0
ELSE
if ADD>=63500&&BLOCK=%00000001 then
stop_f=1
endif
endif
wend

gosub I2C_SlAVE_INIT  ' Set the MCU back to I2C slave mode for further operations
return


SPI_TO_I2C:          ' This program reads the F-RAM on chip and then wirte the data to the
EEPROM
HIGH PORTB.1
stop_f2=0
ADD=0;
ADDR=0;
'FOR ADD=0 TO 64000 STEP 64
WHILE STOP_F2=0
SDI_TRIS=1  'Set up the SDI direction to input when reading SPI memory.
SCK_TRIS=0  'Set up the SCK to output when using SPI transmission
'----------------------Read the four bytes-------------------------------------
FM_CS=0;
TX_BUFF=cont_read
gosub WriteSPI
'------------------Sending three byts of address----------------------------
TX_BUFF=BLOCK;
gosub WriteSPI
TX_BUFF=ADD.byte1
gosub WriteSPI
TX_BUFF=ADD.byte0
gosub WriteSPI
'-----------------Read data into the variables--------------------------------
FOR I=0 to 63
gosub ReadSPI
BUFF3[I]=RX_BUFF
NEXT I
FM_CS=1
'ADDR=ADD;
'SDI_TRIS=0;
'I2CWRITE DPIN,CPIN,CON_B,ADDR,[1,2,3,4,5,6,7,8,9,10];
```

```
SDI_TRIS=0    'Clear the SDI direction values
SCK_TRIS=0
'I2CWRITE
DPIN,CPIN,CON_B,ADDR,[BUFF3[0],BUFF3[1],BUFF3[2],BUFF3[3],BUFF3[4],BUFF3[
5],BUFF3[6],BUFF3[7],BUFF3[8],BUFF3[9],BUFF3[10],BUFF3[11]];
I2CWRITE DPIN,CPIN,CON_B,ADDR,[STR BUFF3\64]
ADDR=ADDR+64;
'PAUSE 10
ADD=ADD+64;

if ADD>=63900&&BLOCK=%00000000 then
BLOCK=%00000001
CON_B=BLOCK1
ADD=0
ADDR=0
ELSE
if ADD>=63900&&BLOCK=%00000001 then
stop_f2=1
endif
endif

WEND
RETURN


'---------------------- I2C Communication subroutine  -------------------------

i2cslave:                              ' I2C slave subroutine
SSPIF = 0                         ' Clear interrupt flag
IF R_W = 1 Then i2crd              ' Read data from us
IF BF = 0 Then i2cexit             ' Nothing in buffer so exit
IF D_A = 1 Then i2cwr              ' Data for us (not address)
IF SSPBUF != I2Caddress Then i2cexit ' Clear the address from the buffer
readcnt = 0                       ' Mark as first read
GoTo i2cexit

i2cwr:                                ' I2C write data to us
datain = SSPBUF                    ' Put buffer data into array
Rxbuffer[Rxbufferindex]=datain
Rxbufferindex=rxbufferindex+1

IF rxbufferindex=RxBufferLEN Then    ' end of buffer transfer
WrData=1
rxbufferindex=0
EndIF
```

```
GoTo i2cexit

i2crd:                              ' I2C read data from us
IF D_A = 0 Then
TxBufferIndex = 0
EndIF

While STAT_BF : Wend               ' loop while buffer is full

Repeat
wcol = 0
SSPBUF = TxBuffer[TxBufferIndex]
Until !wcol
CKP = 1                            ' release clock, allowing read by master
TxBufferIndex = TxBufferIndex + 1    ' increment index
IF TxBufferIndex = TxBufferlen Then  ' all bytes have been tx
TxBufferIndex = 0                ' reset index
EndIF

i2cexit:
Return

'-----------------------------------------------------------------------------
'---------------------- Initialize I2C slave mode ----------------------------
I2C_SLAVE_INIT:
CtPIN = 1              ' SCL must be an input before enabling interrupts
DtPIN = 1
SSPADD = I2Caddress     ' Set our address
SSPCON1 = $36           ' Set to I2C slave with 7-bit address
SSPSTAT = 0
SSPIE = 1
SSPIF = 0
RxBufferIndex = 0
TxBufferIndex = 0
Return

'--------------------------------SPI init-----------------------------------
SPI_INIT:           'Initialzing the SPI communication module
SSPIE=0;
SSPADD=0;
SSPSTAT=%01000000    ' SAMPLE AT THE MIDDLE OF DATA OUTPUT TIME,
TRANSMIT ON IDLE RISING EDGE OF SCK
SSPCON1=%00100000    ' SPI MASTER MODE, CLOCK=Fosc/4  ENABLE
HARDWARE SPI PORT
SSPEN=1             ' ENABLE HARDWARE SPI PORT
FM_CS=1              'FRAM CHIP DISABLED
```

```
FM_CS_TRIS=0        'OUTPUT
SCK_TRIS=0;         'Don't forget to bring the engine next time
SDO=0               'START SPI PIN WITH HIGH
SDO_TRIS=0          'OUTPUT
SDI_TRIS=1          'INPUT

return

'-----------------------------------------------------------------------
WriteSPI:            ' 18cLOCKS
WCOL=0           '1
'Rx_buff=SSPBUF   ' CLEAR THE BUFFER
SSPIF=0          ' CLEAR THE INTERRUP FLAG  1
SSPBUF=TX_BUFF   ' SEND THE BYTE           1
IF(WCOL) THEN RETURN                    ' 8+4
WHILE(!SSPIF)
WEND
SSPIF=0;                     1
RETURN                      '2

ReadSPI:
RX_BUFF=SSPBUF       ' Clear the buffer
SSPIF=0              ' Clear the interrup flag
SSPBUF=0             ' Shift out a dummy byte  What is this?
while(!SSPIF)        ' wait for receive byte
wend
RX_BUFF=SSPBUF              ' Get the byte
return


SPI_ERASE:

' Erase the first block first
FM_CS=0
TX_BUFF=cont_WREN      '1
gosub WriteSPI        '42
FM_CS=1               '1

FM_CS=0               '1
TX_BUFF=cont_write     '1
gosub WriteSPI        '42*14

'-----send three bytes of address-------------
TX_BUFF=$00
gosub WriteSPI
GOSUB WriteSPI
GOSUB WriteSPI
for ADD=0 TO 64000
TX_BUFF=$00
GOSUB WriteSPI
Next ADD
FM_CS=1

'Erase the second block
FM_CS=0
TX_BUFF=cont_WREN      '1
gosub WriteSPI         '42
FM_CS=1                '1

FM_CS=0                '1
TX_BUFF=cont_write     '1
gosub WriteSPI         '42*14

'-----send three bytes of address-------------
TX_BUFF=$01
gosub WriteSPI
TX_BUFF=$00
GOSUB WriteSPI
GOSUB WriteSPI
for ADD=0 TO 64000
TX_BUFF=$00
GOSUB WriteSPI
Next ADD
FM_CS=1
GOSUB I2C_SLAVE_INIT ' Switch back to I2C mode right now.
return

'---------------------------------------------------------------------------------

DISABLE
ISR:
Ticks = Ticks + 1
'IF Ticks < 61 THEN NoUpdate
'
' 1 second has elapsed, now update seconds and if necessary minutes and hours.

'
if ticks=3997 then
ticks=0
Second = Second + 1 ' Update second
IF Second = 60 THEN
Second = 0
```

```
Minute = Minute + 1 ' Update Minute
IF Minute = 60 THEN
Minute = 0
Hour = Hour + 1 ' Update Hour
IF Hour = 24 THEN
Hour = 0
ENDIF
ENDIF
ENDIF
endif
'
' End of time update
'
NoUpdate:
'INTCON.2 = 0 ' Re-enable TMR0 interrupts
TMR2IF=0
Resume
ENABLE ' Re-enable interrupts
END

end    'End of the program
```

2. BIRD Interface box

```
'***************************************************************
'*  Name    : INTERFACE#10                                      *
'*  Author  : [select VIEW...EDITOR OPTIONS]                    *
'*  Notice  : Copyright (c) 2010 [select VIEW...EDITOR OPTIONS] *
'*          : All Rights Reserved                               *
'*  Date    : 8/10/2010                                         *
'*  Version : 1.0                                               *
'*  Notes   : User friendly indications                         *
'*          :                                                   *
'***************************************************************
'------------------------------LCD Definition-----------------------------------
CLEAR
DEFINE OSC 4                                    ; Define LCD connections
DEFINE  LCD_DREG        PORTD           ;
DEFINE  LCD_BITS 4                      ;width of data path
DEFINE  LCD_DBIT 4                      ;data starts on bit 4
DEFINE  LCD_RSREG       PORTE           ;
DEFINE  LCD_RSBIT       0           ;
DEFINE  LCD_EREG        PORTE           ;
DEFINE  LCD_EBIT          1                  ;
'---------------------------------------------------------------------------
'--------------------------------Flags---------------------------------------
Stop_F var byte                       ;The Stop flag for WHIHE structure
RX_F   VAR BYTE                          ;RS232 all bytes received flag
'---------------------------------------------------------------------------
stop_f=0
RX_F=0
'---------------------------------------------------------------------------
'-------------------------Serial Port setting up------------------------------
DEFINE HSER_BAUD 19200
dEFINE HSER_RCSTA 90H
DEFINE HSER_TXSTA 24H
DEFINE HSER_SPBRG 12

'DEFINE HSER_RCSTA 90h ' Enable serial port & continuous receive
'DEFINE HSER_TXSTA 24h ' Enable transmit, BRGH = 1
'DEFINE HSER_SPBRG 51 ' 9600 Baud @ 20MHz, 0,16%
'DEFINE HSER_CLROERR 1 ' Clear overflow automatically
'---------------------------------------------------------------------------



'-------------------------For EEPROM Reading----------------------------------
BLOCK0  CON  %10101000               ;Memory first Block
BLOCK1  CON  %10101010               ;Memory second Block
CON_B   VAR  BYTE                    ;Control byte for EEPROM selection
D_X     VAR  WORD                    ;Variables, for I2C read
D_Y     VAR  WORD
D_Z     VAR  WORD
X       VAR  WORD
Y       VAR  WORD
Z       VAR  WORD
R_X     VAR  WORD
R_Y     VAR  WORD
R_Z     VAR  WORD
P_ADD   VAR  WORD                      ;EEPROM address
D_M     VAR  BYTE                   ;Minute
D_S     VAR  byte                ;Second
D_H     VAR  BYTE                   ;Hour
D_T     var  WORD                  ;Hundredth second
I_I     var  WORD
NOP     VAR BIT
'---------------------------------------------------------------------------
CON_B=BLOCK0      'Selection of memory blocks(For memories larger than 64Kbytes)
P_Add=0           ' Initialzie memory reading address
'---------------------------------------------------------------------------
```

```
'---------------------------Erase VARIABLES-------------------------------------
CLC  var  byte[64] ' Memory erasing array
COUNTER  var  word ' Memory Erasing address index
ADD      var  word ' Index
I        VAR  BYTE ' Index
'-------------------------------------------------------------------------------
'--------------------This Section initialized the array for erasing memory
FOR COUNTER=0 TO 63 STEP 1
CLC[COUNTER]=$00
NEXT COUNTER


'-------------------------------------------------------------------------------



'----------------------------RS232 Initialization-------------------------------
RxData  var BYTE[10]               ; RS232 Data Aray
RxIndex var byte                   ; RS232 Data Index


TRISC = %10111111                  ; set TX (PORTC.6) to out, rest in
'SPBRG = 25                          ; set baud rate to 2400
'RCSTA = %10010000                 ; enable serial port and continuous receive
'TXSTA = %00100000                  ; enable transmit and asynchronous mode
'-------------------------------------------------------------------------------
RxIndex=0                    ; Initialize varibales
RxData=0
'-------------------------------------------------------------------------------
'-------------------------Initialzation of the MCU PORTS----------------------
I2CADDRESS var byte                 ; Slave MCU address
I2CADDRESS=$3                      ; Slave MCU address ini
'ADCON1=7                          ; Set Port A do digital ports

CPIN  VAR  PORTC.3                 ; Define I2C communication pins for
DPIN  VAR  PORTC.4                 ; master mode

LOW PORTE.2                                    ; LCD R/W low (write)
PAUSE 200                                      ; Wait for LCD to start

'OPTION_REG.7 = 0                           ; Enable PORTB pullups to make B4-B7 high
TRISB = %11110000                          ; Make B0-B3 outputs, B4-B7 inputs
'-------------------------------------------------------------------------------


'-----------AD conversion Setup for battery voltage monitoring------------------
Define       ADC_BITS       10      ' Set number of bits in result
Define      ADC_CLOCK           3       ' Set clock source (3=rc)
Define      ADC_SAMPLEUS  50            ' Set sampling time in uS

TRISA  = %11111111          ' Set PORTA to all input
ADCON1 = %10000010  ' Set PORTA analog and right justify result

Bat_V    var word    ' Varibale for battery voltage
Bat_L    VAR BYTE    ' Variable for battery level display
Bat_V=0              ' Initialize the variable

'-----------------------------------------------------------------------------


'******************************* Main Program
*********************************
'*************************************************************************
*****
LOOP:
GOSUB DISPLAY_C

hSERIN [str Rxdata\10] ' This command let the program waits here until the communication
happens
' So the Voltage is not updated until transmission happended or chip-reset

FOR I=0 TO 9          ' THis three lines convert the received ACII code number into actual
decimal numbers
RXDATA[I]=RXDATA[I]-48 '
NEXT I                '

if RxData[9]=3 then             ; LSB byte=03 for accessing memory data
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"INTERFACE-->PBIRD..."
gosub Read_Sent                 ; and send to PC Via RS232

else

if RxData[9]=5 then             ; Memory Erase, it may take 30 seconds...
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"ERASE INTERFACE..."
gosub Erase
ELSE

if RxData[9]=2 THEN        'The frequency is recognized by the six digit of RxData, which is
the thrid digit of
```

```
SELECT  CASE RxData[6]              ' The actual frequency
case 0             'For requency 3050.
RxData[8]=0
RxData[7]=0
case 2
RxData[8]=0
RxData[7]=1
case 4
RxData[8]=0
RxData[7]=3
case 9
RxData[8]=0
RxData[7]=6
case 6
RxData[8]=9
RxData[7]=9
end select
RxData[5]=0
RxData[6]=0
gosub Tx_Sen
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"FREQUENCY SET!"
pause 600
ELSE

IF RxData[9]=1 THEN          'Frequency Select Table
SELECT  CASE RxData[8]
case 2
RxData[8]=2
RxData[7]=0
case 8
RxData[8]=3
RxData[7]=0
case 4
RxData[8]=4
RxData[7]=0
case 1
RxData[8]=7
RxData[7]=0
case 3
RxData[8]=9
RxData[7]=0
case 9
RxData[8]=7
RxData[7]=1
```

```
case 5
RxData[8]=5
RxData[7]=3
case 0
RxData[8]=0
RxData[7]=0
end select
RxData[6]=0
GOSUB Tx_Sen
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"THRESHOLD SET!"
pause 600
else

if RxData[9]=0 then
GOSUB  Tx_Sen
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"SENSOR-->INTERFACE..."
Pause 35000

else
if RxData[9]=9 then
GOSUB  Tx_Sen
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"ERASE SENSOR..."
Pause 6000

else
if RxData[9]=8 then
GOSUB  Tx_Sen
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"SYNCHRONIZATION SET!"
Pause 600

else
GOSUB  Tx_Sen                  ; all others are sent to the sensor
LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
LCDOUT $FE, $C0,"COMMAND SET!"
Pause 600

endif
```

```
endif
endif
ENDIF
endif
endif
ENDIF

GOTO LOOP


'***************************************************************************
*****
'***************************************************************************
*****


'--------------------------------------------------------------------------------

'----------------------------Sub-program for control data display---------------
DISPLAY_C:

LCDOUT $FE, 1
lcdout $FE, $80,"****BIRD CONFIG****"
'LCDOUT $FE, $C0,"C_W=",dec1 RxData[0],dec1 RxData[1],dec1 RxData[2],dec1
RxData[3],dec1 RxData[4],dec1 RxData[5],dec1 RxData[6],dec1 RxData[7],dec1
RxData[8],dec1 RxData[9]
'pause 10
RETURN
'--------------------------------------------------------------------------------




'--------------------------------------------------------------------------------
Tx_Sen:                         ; Sending the control information
LCDOUT $FE, 1                   ; To the sensor board
lcdOUT $FE,$C0,"Sending data"
I2CWRITE DPIN,CPIN,I2CADDRESS,[STR RxData\10],BOGUS ' WIRTE OFFSET TO
SLAVE
return
'--------------------------------------------------------------------------------


'--------------------------------------------------------------------------------
Read_Sent:                          ; Read data from sensor prototype and Serially send them to
PC
```

```
P_ADD=0                         ; Renew the address
con_b=BLOCK0
while stop_f=0
I2CREAD DPIN,CPIN,CON_B,P_ADD,[D_X]  ; Read Channel X
P_ADD=P_ADD+2
'pause 10
I2CREAD DPIN,CPIN,CON_B,P_ADD,[D_Y]  ; Read Channel Y
P_ADD=P_ADD+2
'pause 10
I2CREAD DPIN,CPIN,CON_B,P_ADD,[D_Z]  ; Read Channel Z
P_ADD=P_ADD+2
'pause 10
I2Cread DPIN,CPIN,CON_B,P_ADD,[D_H]
P_ADD=P_ADD+1
'pause 10
I2CREAD DPIN,CPIN,CON_B,P_ADD,[D_M,D_S]  ' Read Minutes and Seconds
P_ADD=P_ADD+2
'pause 10
i2cread dpin,CPIN,con_b,P_ADD,[D_T]      'Read Miliseconds
P_ADD=P_ADD+2
'pause 10
i2cread dpin,CPIN,con_b,P_ADD,[I_I]
P_ADD=P_ADD+5
'pause 10
'Updata address to next dataset's starting address


'IF D_X=0&&D_Y=0&&D_Z=0 THEN
'NOP=0
'ELSE
hserout [dec4 d_x]         ; RS232 Sending out
HSEROUT [" "]
HSEROUT [dec4 D_Y]
HSEROUT [" "]
HSEROUT [dec4 D_Z]
HSEROUT [" "]
HSEROUT [dec4 D_H]
HSEROUT [" "]
HSEROUT [dec4 D_M]
HSEROUT [" "]
HSEROUT [DEC4 D_S]
HSEROUT [" "]
HSEROUT [DEC4 D_T]
HSEROUT [" "]
HSEROUT [DEC4 I_I,10,13]
' ENDIF
```

```
if P_ADD>=63980&&CON_B=BLOCK0 then
CON_B=BLOCK1
P_ADD=0
ELSE
if P_ADD>=63900&&CON_B=BLOCK1 then
stop_f=1
endif
endif

wend

LCDOUT $FE, 1                       ; Clear the LCD
lcdout $FE,$80,"Press key to set"
lcdout $FE,$C0,"Download Finished"

return
'-----------------------------------------------------------------------------
'-----------------------------------------------------------------------------
Erase:                              ; Erase the Memory
ADD=0
FOR COUNTER=0 TO 1999               ; Erase the first section
I2CWRITE DPIN,CPIN,CON_B,ADD,[STR CLC\64]        '
'pause 10
ADD=ADD+64

if ADD>=63900&&CON_B=BLOCK0 then
CON_B=BLOCK1
ADD=0
'COUNTER=0
ELSE
if ADD>=63900&&CON_B=BLOCK1 then
stop_f=1
endif
endif

NEXT COUNTER
LCDOUT $FE, 1
LCDOUT $FE, $80,"Erase Complete"
pause 500
return
'-----------------------------------------------------------------------------
'-----------------------------------------------------------------------------
BOGUS:
LCDOUT $FE,1,"COMMUNICATION ERROR"                    ;I2C COMMAND TIMED
OUT
```

```
PAUSE 500
'-------------------------------------------------------------------------------
GOTO LOOP
'branchl 0,[LOOP]
END                                                    ;End of overall program
```

3. Matlab codes for processing data.

3.1 This program identifies impacts recorded by the NI-DAQ. A threshold can be selected to filter the impact out. The impact can also have certain number of leaders and trailers which can also be selected.

File name: BIRD_IMPACT_IDENTIFICATION

Created: 3.29.2010

Last modified: 7.16.2010

```
% This is the post processing for data impact identification
function BIRD_IMPACT_IDENTIFICATION(Thre,L,T)
%Thre=10;
load('C:\Documents and Settings\pcyu\Desktop\BIRD.txt');
A=BIRD;
%T=A(:,1);
A(:,2)=abs(A(:,2)-1.636204)/0.0022;
ACC=A(:,2);
plot(A(:,1),ACC,'x-');
xlabel('Time, Seconds');
ylabel('Acceleration')
title('Original data')
[m,n]=size(A);
Impact_Pointer=1;
Impact_Index=zeros(10000,3);
i=1;
BIRD_Data=zeros(1000,5);
while(i~=m)
    if A(i,2)>Thre
        F=1;
        %j=0; % pointer for processing inside an impact
        Impact_Index(Impact_Pointer,1)=Impact_Pointer;
        Impact_Index(Impact_Pointer,2)=i;
       while(F==1)
        %B(,:)=A(i,:);
        i=i+1;
        if A(i,2)>Thre
            F=1;
        else F=0;
            Impact_Index(Impact_Pointer,3)=i-1;
        end
       end
        Impact_Pointer=Impact_Pointer+1;
    end
    i=i+1;

end
figure
for i=1:Impact_Pointer-1;
    a=Impact_Index(i,2);
    b=Impact_Index(i,3);
    plot(A(a-L:b+T,1),A(a-L:b+T,2),'x-');
   xlabel('Time, Seconds');
   ylabel('Acceleration')
   title('After applying threshold')
    hold on;
    BIRD_Data(i+1,1)=i; %Impact_Data one, impact index
    BIRD_Data(i+1,2)=A(a-L,1); %Impact_Data two,impact starting time
    BIRD_Data(i+1,3)=9.8*trapz(A(a-L:b+T,1),A(a-L:b+T,2));% Numerical Integration to
velocity change
    BIRD_Data(i+1,4)=max(A(a-L:b+T,2));% Impact Peak G
    BIRD_Data(i+1,5)=A(b+T,1)-A(a-L,1);% Impact Duration in MS

end
% BIRD_Data(1,:)=['Index','elapse time','Vel Ch','Peak g','Duration'];
%BIRD_Data(1,:)=[''];
figure;
plot(BIRD_Data(2:Impact_Pointer,3),BIRD_Data(2:Impact_Pointer,4),'*')
    xlabel('Velocity change(m/s)');
    ylabel('Peak G')
    title('Peak G VS Velocity Change')
xlswrite('BIRD',BIRD_Data(1:Impact_Pointer-1,:));
BIRD_Data
```

3.2 Index streamline program

File name: Index_Modify

```
load('C:\Documents and Settings\pcyu\Desktop\BIRD.txt');
A=BIRD;                      %Copy Data array to A
[m n]=size(A);               %Get the size of the array
% Imp_Num=max(A(:,8))-min(A(:,8))+1       %Get the number of Impacts

% Index processing
```

```
Raw_Index=A(:,8);
New_Index=zeros(m,1);
Index_Min=min(Raw_Index);
for i=1:m
    New_Index(i)=Raw_Index(i)-Index_Min;
end
Index_F=New_Index;
for i=1:m-1
    if New_Index(i+1)-New_Index(i)>=1
        for j=(i+1):m
            Index_F(j)=New_Index(j)-(New_Index(i+1)-New_Index(i))+1;
             end
    end
     New_Index=Index_F;
end
BIRD(:,8)=Index_F;
xlswrite('SWAY_L_S.xlsx',BIRD,'8-25R');
```

Appendix B

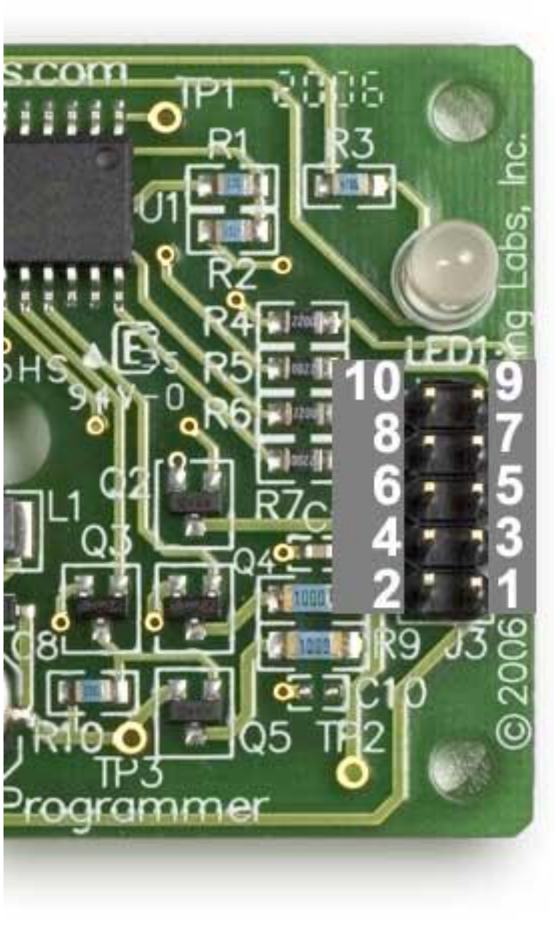
In circuit programming of the BIRD sensor

Tools and equipments:

1. Hardware: ICSP programmer, Wires, USB cable
2. Software: Micro Studio Compiler, Melabs Programmer

Steps(Cited partially from MELABS.COM):

To start in circuit programming, the hardware connection needs to correctly connected.



PIN#	SIGNAL
1	+5V always on (It is not recommended that you use this supply to power your target board. Current capability is limited.)
2	Programming Voltage (Vpp) - Connect to MCLR/Vpp pin on target device.
3	Reserved - do not connect
4	Switched Vdd (This pin should not be used to power your target board. It is intended to power a target PICmicro only when one of our programming adapters is used.)
5	Reserved - do not connect
6	Programming Data - Connect to PGD or ICSPDAT on the target PICmicro
7	Reserved - do not connect
8	Programming Clock - Connect to PGC or ICSPCLK on the target PIC MCU.
9, 10	Ground - A ground connection to your target board is required.

1.1 If possible, dedicate the Programming Clock and Data pins to ICSP. If you must also use these pins for other purposes on your board, some thought should be given to the type of hardware that is connected. The best case is to use the programming pins for normally-open, pushbutton inputs. As long as you don't push the buttons while programming, the switches won't affect things at all.

If loads are connected to the clock and data lines, they must not interact with the clock and data signals. Capacitive loads will cause problems. High-impedance loads are usually ok. LEDs with current-limiting resistors are less desirable, but usually don't cause problems.

You should always prototype and test the ICSP connections before sending your PCB design out for fabrication.

1.2 The MCLR pin on the PIC will be driven to approximately 13V during programming. This raises two concerns. The first is that your circuit must allow 13V on the pin. Don't connect the pin directly to the Vdd rail. The second concern is that the rest of your circuit may need to be protected from this voltage. If 13V on the MCLR pin might put at risk other components on your board, use a diode in series with the RESET pullup to keep the 13V off of the Vdd line.

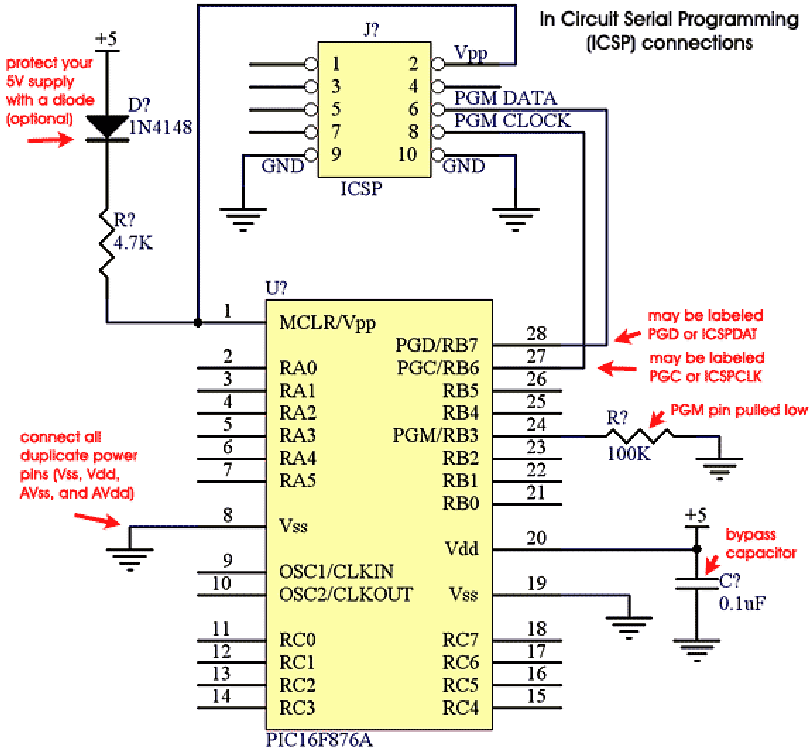
1.3 Configure MCLR as a RESET pin if possible. The programmer will attempt to reset the target device at the beginning of the programming process. If the MCLR pin is configured as an input, the programmer may have problems putting the chip into program-mode. This is especially problematic if the target has been previously programmed and is using the programming clock or data lines as outputs.

This can make reprogramming much easier to accomplish when the programmer has no means to reset the target device.

1.4 Make sure the "low-voltage programming" pin is pulled to ground. Low-voltage programming is not used by melabs programmers and should always be disabled in the configuration settings. Even so, the low-voltage programming pin (labeled PGM or LVP) should be pulled to ground when programming. If it is left floating or in a high condition, it may interfere with the programming process. We use a 100K resistor to pull the pin low when designing a board for ICSP. For most 16F series parts, RB3 doubles as the PGM pin. On 18F series, it is usually RB5 that needs to be low. Notable exceptions are the 16F62x and 16F64x family, which need RB4 to be pulled low.

1.5 Always connect every available power pin to the appropriate power or ground source. Many PIC devices have multiple pins that are labeled Vdd, Vss, AVdd, and AVss. If even one of these pins is left unconnected, it may result in programming errors. This applies to analog supply pins even if you don't plan to use the analog functions.

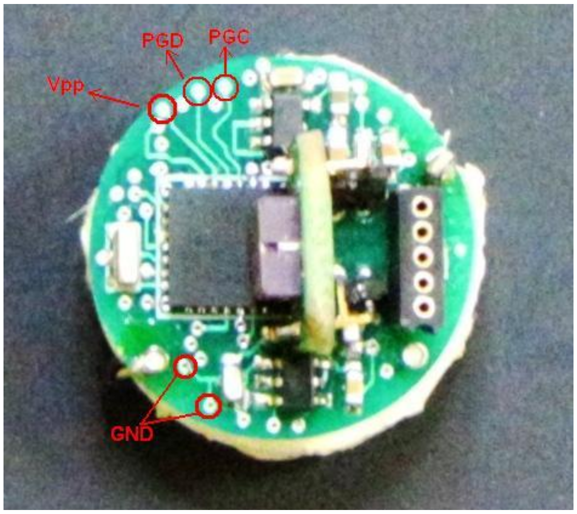
The following schematic shows a typical ICSP connection for the PIC16F876A


In Circuit Serial Programming (ICSP) connections
+5
protect your 5V supply with a diode (optional)
D? 1N4148
R? 4.7K
J?
1 3 5 7 9
2 4 6 8 10
Vpp
PGM DATA
PGM CLOCK
GND
GND
ICSP
U?
MCLR/Vpp
PGD/RB7
PGC/RB6
RA0 RA1 RA2 RA3 RA4 RA5
RB5 RB4 PGM/RB3 RB2 RB1 RB0
Vss
Vdd
OSC1/CLKIN
OSC2/CLKOUT
Vss
RC0 RC1 RC2 RC3
RC7 RC6 RC5 RC4
PIC16F876A
may be labeled PGD or ICSPDAT
may be labeled PGC or ICSPCLK
R? 100K
PGM pin pulled low
connect all duplicate power pins (Vss, Vdd, AVss, and AVdd)
+5
bypass capacitor
C? 0.1uF

1. The BIRD sensor

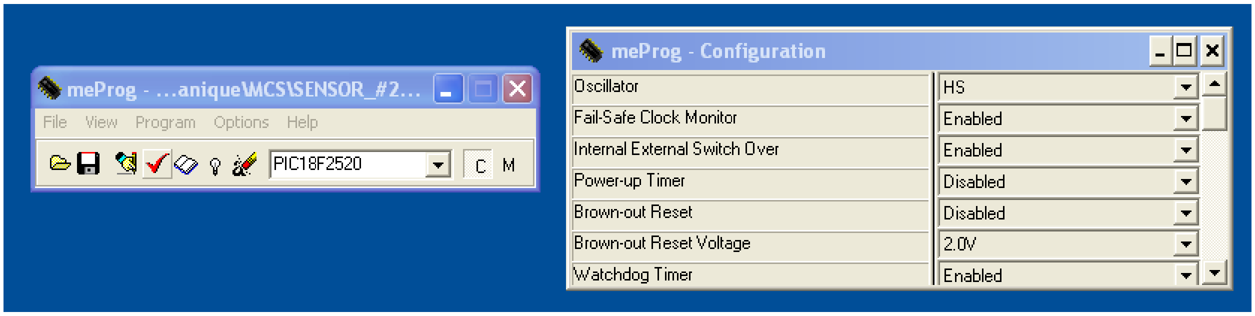
To perform the ICSP for the BIRD sensor, as shown in the following figure.

The four pins that have been circled out using red mark needs to be connected with the ICSP programmer’s respective pins. Power supply to the sensor board is also needed, so, the sensor battery needs to be recharged before in circuit programming.



2. Programming

After correctly connect the BIRD sensor with the ICSP. You can compile the target program using the MicroCode studio. After the melabs Programmer pops out, you need to setup the mode to the following manner: Select HS in the Oscillator mode, and disable the Brown-out Reset, leave all others to default settings.



PIC18F2520

meProg - Configuration

Oscillator	HS
Fail-Safe Clock Monitor	Enabled
Internal External Switch Over	Enabled
Power-up Timer	Disabled
Brown-out Reset	Disabled
Brown-out Reset Voltage	2.0V
Watchdog Timer	Enabled

Then click program command to start the in circuit programming.

Appendix C

BIRD commands list

Data Bits									Control Bit	Description
Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
X	X	X	X	X	X	X	X	X	0	Download sensor data to interface board
X	X	X	X	X	X	a	b	c	1	Theshold Setup: Threshold=(ax100+bx10+c)x1.5(G)
X	X	X	X	X	X	X	a	b	2	Frequency Setup(Hz) ab=99,F=682Hz; ab=60,F=998Hz; ab=30,F=1480Hz,ab=20,F=1760Hz; ab=10,F=2210Hz; ab=00,F=3050Hz
X	X	X	X	X	X	X	X	X	3	Upload data to PC
X	X	X	X	X	X	X	X	X	4	Enable BIRD sensor for data collection
X	X	X	X	X	X	X	X	X	5	Erase interface memory data
X	X	X	X	X	X	X	a	b	6	Power Control:a=1, sleep mode a=0,b=0,standy by, a=0,b=1,active mode
X	X	X	X	X	X	X	X	X	7	
H	H	M	M	S	S	m	m	m	8	Synchronization: H=hour, M=Minutes, S=Seconds,m=miliseconds
X	X	X	X	X	X	X	X	X	9	Erase Sensor Memory

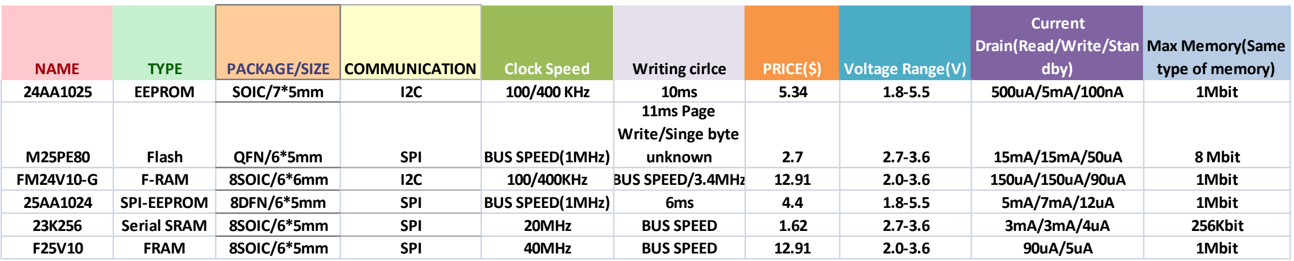
Appendix D

List of MEMS Accelerometers

	manufacture	Part NO.	# axes	range	sensitivity	accuracy%	bandwidth/sampling frequency KHz	voltage supply	current supply full/stand by	package size	price $
1	Analog Device										
	High g accelerometers	ADX278 (analog)	X/Y	±70/±35 or ±50/±50	27 mV/g	-5	0.4	4.75-5.25	2.2mA	5x5x2 mm	7.95
		ADXL78 (analog)	X	±50 or ±70	28 mV/g	-5	0.4	4.75-5.26	1.3mA	5x5 mm	5.66
		ADXL001(analog)	X	±70,±250,±500	3.3V:16/4.4/2.2 5V:24.2/6.7/3.3	0.2% Full scale Non-lineraity	22KHz	3.3, 5V	2.5mA	5x5 mm	40.43
		ADIS16204 (digital SPI)	X/Y	±70, ±37	17.13mg/LSB		0.4	3-3.6	Normal:12mA Fast : 37mA Sleep : 150uA	9.35×9.20 mm	21.2
		1. no triaxis sensor with high g;	2. highest range is 70g	3. package is small;	4. it's possible to combine ADX278 with ADXL78 using all 50g						
2	FreeScale	MMA3202(analog)	X/Y	±100/±50	X-20mV/g Y-40mV/g	Nonlineraity ±1	0.4	4.75-5.25	8mA	13×10 mm	6.8
		MMA1210(analog)	Z	±100	Z-20mV/g	Nonlineraity±1	0.4	4.75-5.25	5mA	10.45×10.67mm	4.98
		MMA2301(analog)	X	±200	X-10mV/g	Nonlineraity±1	0.4	4.75-5.25	3.0-6.0mA	10.45×10.67mm	8.45
		MMA1212(analog)	Z	±200	X-10mV/g	Nonlineraity±1	0.4	4.75-5.25	3.0-6.0mA	10.45×10.67mm	8.68
		MMA1200(analog)	Z	±250	X-8mV/g	Nonlineraity±1	0.4	4.75-5.25	3.0-6.0mA	10.45×10.67mm	8.68
	Medium g and high g accelerometers, no tri-axial avaiable. Relative bigger package size with with typically a couple of mA current supply. Most are analog sensors with 5 Volts supply. Possibly we could use three chips to combine together as a tri-axes design.										
3	Measurement Specialties, Inc	ACH-04-08-05(analog)	X/Y/Z	±250/±250/±250g	1.8mV/g	Nonlineraity±1	0.5-5	5	22uA/4uA	11.43×10.87 mm	159
		3038	X	50-6000g dynamic range	2.0-0.1mV/g @10Vdc Excitation voltage	Nonlineraity±0.5-±2	±50g: 0-1000Hz ±100g: 0-1500Hz; ±200:0-2000Hz ±500:0-4000Hz	2-10V	NA	7.6×7.6×3.3	
	Notes: Piezo-electronic embedded sensors This specific chips needs 10 weeks in advance for order and relative costive. The 3038 series provide wide selection of single axis accelerometers with wide dynamic range and frequency response. So there is no frequency and acceleration range.										
4	VTI	SCA3100-D07(digi	X/Y/Z	±2,±6	70mg/LSB	±3	0.055	3-3.6	5mA/50uA	7.6×3.3×8.6	61.24
		SCA3000-E05	X/Y/Z	±18	160coung/g	±11	0.06	2.35-2.7	120uA	7×7×1.8 mm	23.33
	Note: good features in providing tri-axes Medium g acceleration										
5	STMicro-electronics	LIS302DL(Digital SPI)	X/Y/Z	±2±8	18,72 mg/count	Nonlinearity±0.5	0.1-0.4	2.16-3.6	0.3-0.4mA	3×5×0.92 mn	10.39
		LIS331HH(I2C SPI)	X/Y/Z	±6±12±24	3,6,12 mg/count	Nonlinearity±0.5	0.05,0.1,0.4,1	2.16-3.6	250uA/1uA	3×3×1 mm	6.6
	Note: Low g(±2±4±8), or Medium g(±24) MEMS, small package, no high G sensor available										
6	Kionics										
		KXSD-1026(Digita	X/Y/Z	±2,±4,±6,±8	819,410.273,105 count/g	XY:±1.1, Z:±0.6	0.5-2	1.8-.36	220uA/0.3uA	3×3×0.9	
		KXD94(Analog)	X/Y/Z	±15	200mv/g	XY:±1.6, Z:±0.2	0.8	2.5-5.25	1.2mA/5uA	5×5×1.2	
	Note: Low g tri-axes MEMS accelerometers, no medium or high g selections										
7	Bosch Sensor	BMA020(I2C,SPI)	X/Y/Z	± 2g, 4g, 8g				20.-3.6	200/1 uA	3×3×0.9	6
		SMB48x/49x	X	±480g		7%	426Hz	5-11V	6mA	N/A	
	Bosch provide same featurs in designing tri-axes accelerometers with low g, MEMS miniature package.No high g sensors										
8	Silicon Designs										
	Silicon Designs provides good acceleration ranges upto 400g. Relative small package size with high-power consumption(2-8ms).	Model 1010(Digital Sequential output)	Z	±2,±5,±10,±25,±50,±200g	62.5/25/12.5/5/2.5/1.25/0.625(mV/g)	2%~4%	400/600/1000/1400/1600/1800/2000	5	2mA	9.0×9.0×2.9 (mm)	
		Model 1210(±4V Differential Output/0.5-4.5V Single Ended Output)	Z	±5,±10,±25,±50,±100,±200,±400	800/400/160/80/40/20/10(mV/g)	2%~4%	400/600/1000/1500/2000/2500/3500	5	7mA	9.0×9.0×2.9 (mm)	

Appendix E

List of serial memories



NAME	TYPE	PACKAGE/SIZE	COMMUNICATION	Clock Speed	Writing cirlce	PRICE($)	Voltage Range(V)	Current Drain(Read/Write/Standby)	Max Memory(Same type of memory)
24AA1025	EEPROM	SOIC/7*5mm	I2C	100/400 KHz	10ms	5.34	1.8-5.5	500uA/5mA/100nA	1Mbit
M25PE80	Flash	QFN/6*5mm	SPI	BUS SPEED(1MHz)	11ms Page Write/Singe byte unknown	2.7	2.7-3.6	15mA/15mA/50uA	8 Mbit
FM24V10-G	F-RAM	8SOIC/6*6mm	I2C	100/400KHz	BUS SPEED/3.4MHz	12.91	2.0-3.6	150uA/150uA/90uA	1Mbit
25AA1024	SPI-EEPROM	8DFN/6*5mm	SPI	BUS SPEED(1MHz)	6ms	4.4	1.8-5.5	5mA/7mA/12uA	1Mbit
23K256	Serial SRAM	8SOIC/6*5mm	SPI	20MHz	BUS SPEED	1.62	2.7-3.6	3mA/3mA/4uA	256Kbit
F25V10	FRAM	8SOIC/6*5mm	SPI	40MHz	BUS SPEED	12.91	2.0-3.6	90uA/5uA	1Mbit