An All day wrist motion tracking device using low power consumption components

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Surya Prakash Sharma
August 2014

Accepted by:
Dr. Adam W. Hoover, Co-chair
Dr. John N. Gowdy, Co-chair
Dr. Richar R. Brooks
Dr. Eric R. Muth

Acknowledgments

To Fire, Water, Earth and Wind. Because without Captain Planet, we wouldn't really have anything to eat.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis considers the problem of tracking human wrist movements during free living. This is motivated by the fact that wrist and hand movements can be used detect periods of eating [1] and then further track the amount of food that is being ingested by a person [2]. Eating can occur at any time of the day, and it very hard to monitor the activity as such. We propose a solution which uses a wrist watch like device to monitor how the wrist is moving throughout the day, and store it on a memory chip.

1.1 Motivation

Obesity is an increasing problem. CDC (Centers for Disease Control and Prevention) reports mention one — third of the adult American population as obese, while sixty nine percent of the population is considered overweight. The CDC defines individuals with a Body Mass Index of twenty five and above as overweight [3]. This number is rising in children too, with eighteen percent children between the ages of 12 — 18 years considered obese. Modern advances have helped our minds work on autopilot. Not only is it easy to order food, but consuming food while watching television or using a computer makes it very easy to consume a large amount of calories quickly without noticing it happen. when a group of individuals was asked to count the number of bites taken over a 24 hour period, 40% lost count or forgot entirely [4]. The abundance of fast food chains, high calorie food, and soda dispensers makes it very easy for a person to consume a large amount of calories without realizing that they are doing so. Children who ate fast food, compared with those who did

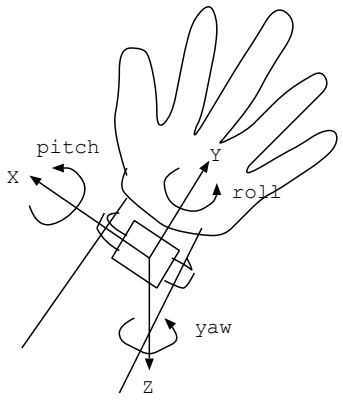


Figure 1.1: A wrist watch shaped device showing the different axis being recorded.

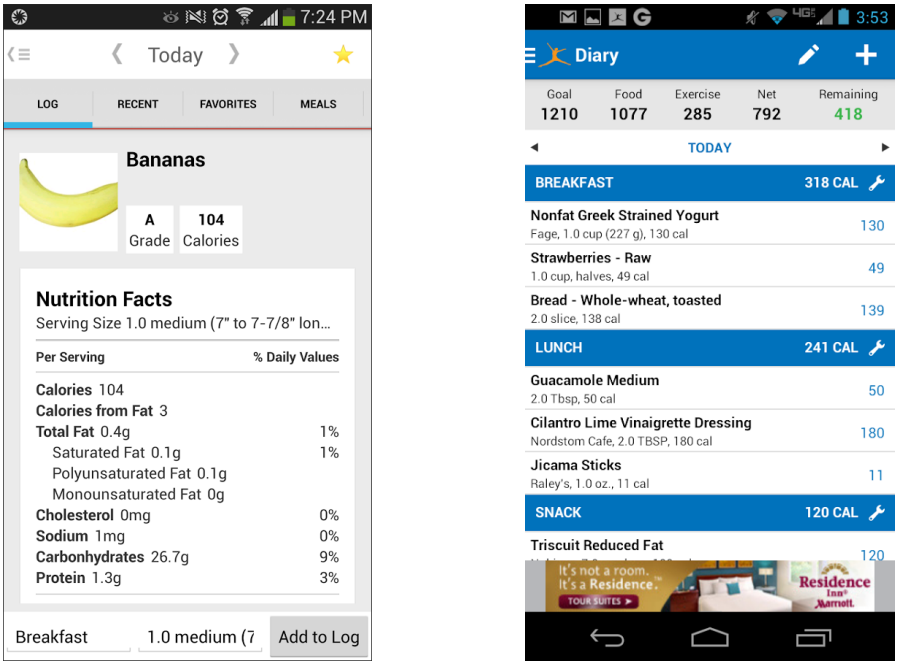
not, consumed an average of 187 kcal more every day [5]. Eating slowly is shown to help decrease the amount of calories consumed during a meal [6], however it is not an easy change to make in our busy lives.

1.2 Fitness Tracking

With the increase in smart phone sales, we see a huge rise in small applications that help track our daily activities. Applications like CalorieCount[1] and MyFintessPal[2] allow users to track their daily intake of food and also track other activities like exercise and sleep. However, this data was still largely dependent on the user's ability to record these events. Smaller sensors and components have allowed wearable devices to enter the market. These devices are a part of clothing or accessories that contain a microcomputer to perform electronic functions, and can have a battery life of 24 to 48 hours.

Fitness trackers in the market currently offer a range of services. The Fitbit[3] and Jawbone[4] series of sensors allow for exercise and sleep monitoring using wrist movements. This data is then synchronized with a computer or a mobile phone, and can be viewed graphically by the user. Another consumer electronic segment that has now emerged is smart watches. These watches contain electronic components that make them functionally similar to a mobile phone, and are small enough to wear on the wrist. They contain low power displays and have a host of power saving techniques

[1] Allows users to name and record calories consumed. Users can pick from a database or enter their own values.
[2] Allows users to pick various foods or exercises from a database or enter their own data.
[3] http://www.fitbit.com
[4] http://www.jawbone.com



(a) Calorie Counter (b) MyFitnessPal

Figure 1.2: Screenshots of two famous Fitness Tracking Software

Figure 1.3: The Fitbit flex (left) and the Jawbone Up activity trackers

to have a long battery life. However most of the wearable devices on are limited to counting the number of calories expended and thus not many options exist for counting the number of calories consumed.

1.3 Wearable Wrist Motion Tracker

Work done in [7] shows a device that is capable of detecting bites by monitoring wrist movement We are motivated by the problem of monitoring calorie consumption by creating a wrist watch like device that automates the process. Since the device is to be worn by the user everyday, it needs to have a small size. To avoid having to recharge often, or risking the battery dying during a meal, the device also needs to have a good battery life[5]. Our device records raw inertial movement data (Rotation and Acceleration), and stores this in a memory chip. The configuration of the various axes is displayed in Figure 1.1. The data can later be accessed later using a computer and can processed by algorithms shown in [2] to show when a bite of food has been eaten.

1.4 Previous Work

1.5 Shimmer

[5]Requiring a recharge not more than once a day

Chapter 2

A new device to record wrist movement in free living humans

This section covers the actual methods and techniques used to create our Wrist Motion Activity tracker. We discuss the various hardware techniques used to create the device (2.1). This includes different sensors available, picking the right one for our purpose and then prototyping a device. Note that since the wrist activity motion tracker is to be mounted on the wrist, it needs to be a small device that can be worn for a long interval[1] without discomfort to the user. This means that it also needs to have a long battery life, which is dictated by two factors: The size of the battery, and the current consumption of the parts on the device.

Then we discuss the software techniques and code used to create the device or the software to interact with it that runs on a desktop computer. Two different software were used to analyze the data from the device. This is briefly discussed in the Software section (2.1.11).

Our ultimate aim to count the number of bites in each meal has made us look towards the rotational and linear movements of the wrist. This in turn has led us to consider linear acceleration and angular rotation data which is sensed by an accelerometer and a gyroscope respectively.

[1] At least 16 hours

2.1 Hardware

To create the prototype device, different prototype breakout boards were used (2.1.5). These boards allow rapid prototyping by hooking up components using wires. This reduces any time spent to solder an Integrated Circuit to the PCB or to understand its typical application circuit components. Next, we consider the different devices used to create our wrist motion activity tracker. We would need a power source in the form of a battery (2.1.11), sensors to measure acceleration (2.1.2) and angular velocity(2.1.3), a memory chip (2.1.6) to store data from the sensors and a microcontroller (2.1.7)to control this system and act as a communication channel between the different devices. To connect this device to a computer, we would need a translator. We use a USB to UART bridge (2.1.10) to accomplish this.

With all these components available, we would have to connect them together correctly, which is explaining in the Circuit Design section(2.1.11). Once the design has been tested, it is laid out as a PCB design using EAGLE PCB, and then fabricated. This PCB would then need the IC's soldered to it, and we discuss the two methods used to do so (2.1.12).

We recap that we have two strong requirements for our device, and all hardware decisions must consider these:

- Device Size
- Device Power Consumption

With this in mind, we look at the different hardware in our device in the next section.

2.1.1 Sensors

Since we want to log acceleration and rotation data, we looked at the different sensors available in the market and their sizes. The clear winner was a category of sensors called MEMS sensors. MicroElectroMechcanicalSystems based sensors are small devices ranging from about 4 mm^2 to 25 mm^2. Our device design uses an MEMS gyroscope and accelerometer in a single electronic chip to sense movements. This chips may also be called the IMU or the Inertial Measurement Unit. The orientation and axes for this chip is shown in figure 2.1.

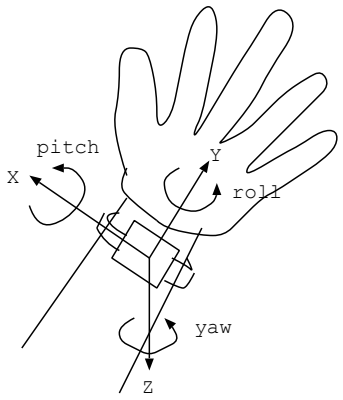


Figure 2.1: Orientation of axes in the IMU when mounted on the wrist

2.1.2 Accelerometers

Accelerometers are devices that sense linear acceleration. In the case of our device we would like to measure the linear acceleration in all three Cartesian co - ordinate axes, X, Y, and Z. This will allow us to track the movement of the wrist in free living humans as their hand moves around in free space. Figure 2.1 shows how the axis of the sensor are oriented for our device.

To understand how an accelerometer works, we imagine a system devised of a spring connected to a fixed surface at one end as shown in figure 2.2, and an object with a mass M at the other end. Since springs are governed by Hooke's Law, we can use that to explain its behavior. Hooke's law states that to extend or compress a spring by distance X, the force required is proportional to X, or mathematically as given in equation 2.1.

$$F = -kX \tag{2.1}$$

where F is the force required, X is the distance the spring is extended or compressed from its equilibrium position, and k is the spring constant for a particular spring.

We also have Newton's Second Law of Motion which states that the force applied on a body with constant mass produces a proportional acceleration.Mathematically, Newton's Second Law of motion can be expressed as given in equation 2.2.

$$F = ma \tag{2.2}$$

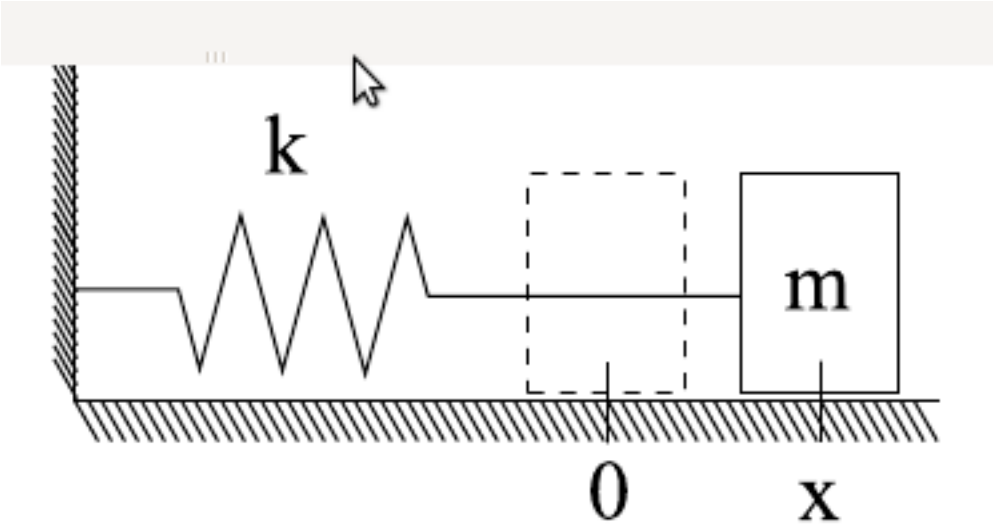


Figure 2.2: Example of a spring system

where F is the force, m is the mass of the body, and a is the acceleration produced. Combining the two equations, we can solve for acceleration using the result in equation 2.3.

$$a = \frac{-kx}{m} \tag{2.3}$$

However, consider the fact that the spring constant k and mass m are constant in the equation above. This means that acceleration a is inversely proportional to the displacement x. Knowing this, new MEMS accelerometers are very simple systems consisting of small mechanical parts that measure the displacement and output an acceleration based on this.

We must remember that since the sensor is measuring force, that gravity is always acting on the device, and due to this, a reading of 1g will always be read on the Z axis for the "earth" system. If there are no rotational movements, we know which direction gravity is acting in, and can subtract this from the readings obtained from an accelerometer.

2.1.3 Gyroscopes

Gyroscopes are devices used to measure angular movement. In our case, we use MEMS gyroscopes to measure angular velocity in three axes, X, Y and Z (as shown in figure 2.1). How a

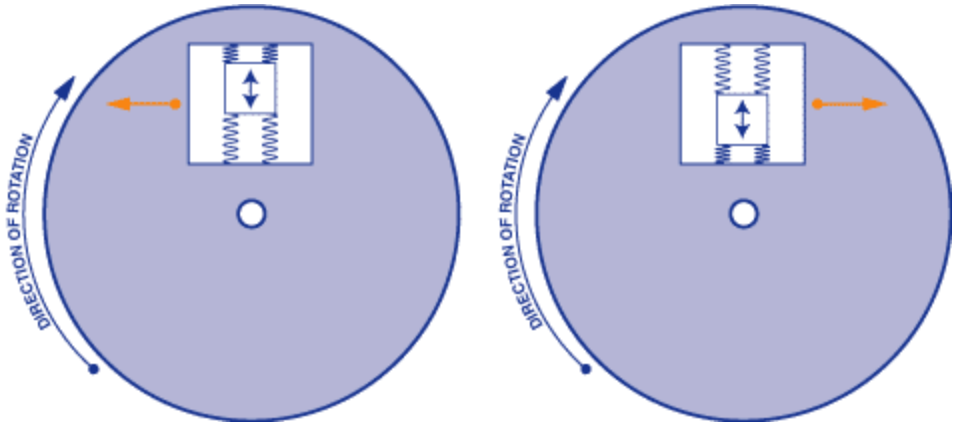


Figure 2.3: Internal Working of a Gyroscope

gyroscope works in explained in [8]. We will review the same here.

An MEMS gyroscope contains a small mass that oscillates inside it as shown in figure 2.3. When the gyroscope rotates, the oscillating mass is displaced between two oscillations. This change in location is converted into a very low current, that is then amplified (and converted to a voltage) by a microcontroller inside the gyroscope integrated chip.

2.1.4 Accelerometer and Gyroscope Device selection

There are a variety of MEMS accelerometers and gyroscopes available in the market with different specifications. Our purpose was to pick the most suitable device for logging wrist movement data. We will first look at the different parameter's that we have for MEMS devices. These parameter's are referenced from Analog Devices' website[2] :

Output

The output of the sensor can be analog or digital. For an analog device, the output is a voltage corresponding to the detected measurement. This means that an acceleration between -A_{max} to +A_{max} would be output as a voltage between V_{dd} and V_{cc}. Each output would have its own pin to output a signal. A digital device on the other hand, would use some kind of communication protocol. The device usually has registers or a memory stack that can be read using a protocol like I^2C (also known as TWI) or SPI. Using this protocol, a master device (microcontroller) can poll the slave (accelerometer) to read its memory.

[2]Analog Devices - Accelerometer Specifications - Quick Definitions. Last accessed October 2014

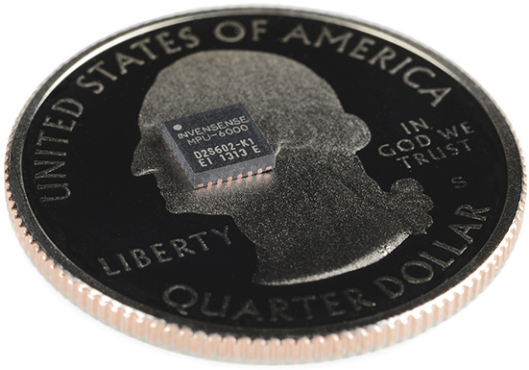


Figure 2.4: The MPU6000 IMU lying on an American Quarter for size comparison.

Output data Rate

This defines the rate at which data is sampled and then output by the device. In analog accelerometers, this represents how often an accelerometer updates its output voltage in response to the measured acceleration. In digital accelerometers, this represents how often the register storing the value of the acceleration is updated. Our required data rate is of 15Hz for sufficiently accurate monitoring of wrist movements.

Package type and Size

Since the device is mounted on the wrist, space and volume is at a premium. Most MEMS devices are catered to the consumer electronics market, and used in small devices like mobile phones and smart watches. Due to this, there is constant innovation in the market with regards to the size of the integrated chip's footprint. Packages like LGA (Land Grid Array) and QFN (Quad Flat No - Lead) are used. These packages usually lack pins and are surface mounted to reduce the footprint on a PCB. An example of this is the IMU that we used in our device, the MPU 6000 produced by Invensense Inc. An image from Sparkfun.com [9] (see figure 2.4) shows the size of the chip compared to a quarter.

Measurement Range

This defines the range of measurement supported by an IMU's output specifications. For an accelerometer, the range is usually defined in g's, where 1g = 9.8 m/s^2. This is the greatest amount of acceleration that can be measured and accurately represent as an output. A larger acceleration maybe incorrect or capped at the maximum value. For example, the MPU-6000 has four modes with measurement ranges from ±2g to ±16g. The measurement is not the breaking point for the device. A higher acceleration of 17g may not break the accelerometer, but will not be reported correctly. This value is actually defined as the Absolute Maximum Acceleration.

Human motions are typically contained within 2g's of acceleration. With this fact, our options for accelerometers is fairly large considering that most MEMS accelerometers aimed at consumer electronics will have a measurement range of 2g's.

Current Consumption

The power drained by the device is a huge factor for us. MEMS accelerometers are fairly low power devices. The LIS344ALH accelerometer used by Drennan in [7] typically consumed 650 μA of current when operating and a maximum of 5μA when in sleep mode. Gyroscopes on the other hand tend to consume a very high amount of current. [7] used two gyroscopes, each consuming 6.8mA. This means that on a standard coin cell battery that has a size of 90mAh, the gyroscopes would run for about 7 hours, a very short lifetime.

Number of Axes

As MEMS technology has improved, the sensors abilities have improved. Early accelerometers would sense acceleration across a single axis. As of now, most accelerometers and gyroscopes can measure across all three axes. Inertial Measurement Unit's are available that pack not only an accelerometer and a gyroscope, but also a magnetometer for more accurate measurements. This greatly reduces the current consumption against a scenario where multiple devices are being used. We will not be using a magnetometer for our work.

With these parameters, we compared a few of the options available in the market in September 2013. Table 2.1 shows the different product names versus their important parameters. The first three products are the ones used in [7]

Name	Description	Output	Package	Current Consumption	Noise	Full Scale Range
LPR410AL	2 Axis Gyro (pitch, roll)	Analog	LGA 4X5X1.1mm	6.8mA	0.014 (°/s/ Hz)	100 °/sec
LPY410AL	2 Axis Gyro (pitch, yaw)	Analog	LGA 4X5X1.1mm	6.8mA	0.014 (°/s/ Hz)	100 °/sec
LPR344ALH	3 Axis Accelerometer (X,Y,Z)	Analog	LGA 4X4X1.5mm	680uA	50 (ug/ Hz)	2/6 g
LGD20H Series	3 Axis Gyro (yaw,pitch,roll)	Digital	LGA 3X3X1.1mm	6.1 - 6.4mA	0.03 (°/s/ Hz)	245/500/2000 °/sec
L3G462A	3 Axis Gyro (yaw,pitch,roll)	Analog	LGA 4X4X1.1mm	6.9mA	0.17 (°/s/ Hz)	625 °/sec
LSM330	6 Axis (X,Y,Z,y,p,r)	Digital	TFLGA 3.5X3X1mm	6.1mA	Gyro: 0.03°/Hz	[2 g/4 g/6 g/8 g/16 g] [250/500/2000 °/sec]
Invensense MPU-61NX	6 Axis (X,Y,Z,y,p,r)	Digital	QFN (4X4X0.9)	3.8mA	Gyro: 0.005°/Hz	[2g, 4g, 8g,16g] [56/ 113/225/ 450 °/sec]
Invensense MPU-6XX0	6 Axis (X,Y,Z,y,p,r)	Digital	QFN (4X4X0.9)	3.8mA	Gyro: 0.005°/Hz	[2g, 4g,8g,16g] [250/500/1000/2000°/sec]
Invensense MPU-91NX	9 Axis (6 Axis + magnetometer)	Digital	QFN (4x4x1mm)	3.8mA	Gyro: 0.005°/Hz	[2g, 4g,8g,16g] [250/500/1000/2000°/sec]

Table 2.1: Comparison of Various movement sensors available in the market.

Figure 2.5: A breakout board containing the MPU 6050 chip. Quarter shown for comparison. Image from Sparkfun.com

Based on these different parameters and comparing with [7], we have selected the Invensense MPU-6000. The device contains both, a 3-axis accelerometer and a 3-axis gyroscope. It features three 16-bit analog-to-digital converters each for both these devices to digitize the sensor outputs. This data is stored in internal registers that can be accessed using I^2C at 100kHz or using SPI at 1Mhz. An on-chip buffer allows storing bursts of readings, which can be read in bursts by a master device. The device operates at a voltage of 2.375V to 3.46V, which would mean operation on most Lithium Ion battery. An operating current of 3.8mA means that the device can operate for roughly 24 hours on a standard coin cell battery. This makes the MPU 6000 very suitable for our requirements.

2.1.5 Breakouts

Newer integrated chips are packaged very densely, and have very small pins to connect to a PCB. With surface mount technology, it is possible to solder these devices to PCB's, however it is very hard for hobbyists to do so, because of the lack of expensive equipment. Electronic part retailers provide products in the form of Breakout boards, which means that they provide PCB's with the integrated chip soldered. These PCB's have pins to connect to that can be used conveniently. In effect, a breakout board converts pin - less packages like BGA (Ball grid array) and LGA (Land grid array) to DIP (Dual inline package).

These breakout boards are easier to prototype with, and allow modular creation of a complete circuit. Although we have selected the MPU 6000, we still have to work it into our circuit and ensure that it will work in our device. To do so, conventional methods would require us to first solder this IC to a PCB containing the other components, and then programming a microcontroller to read from the sensor. However, PCB fabrication and soldering can be expensive, and are not a viable option during the prototype period. PCB manufacture can also take a considerable amount of time (up to a month), and is not a feasible strategy to work with. To prototype and test the sensor without fabricating a complete PCB, we decided to use a Breakout board containing the sensor. Figure 2.5 shows an example of a Breakout board. The image was sourced from [10].

2.1.6 Memory

The information sensed by the sensors has to be logged, and then processed by a computer. We could either store this information or transmit it wirelessly as it is gathered. For a system planned to read sensors at 15Hz, and logging data for 24 hours, we estimate the amount of data as shown in Equation 2.4

$$Data = 20\ hours * \frac{60\ minutes}{hour} * \frac{60\ seconds}{minute} * \frac{15\ polls}{second} * \frac{6\ sensors}{poll} * \frac{1\ byte}{sensor\ poll} \tag{2.4}$$

$$Data = 6,480,000\ Bytes = 6,328.125\ KiloBytes \tag{2.5}$$

We do not consider wireless transmission in our case. The device would have to transmit data to another device which it would be paired to. The paired device would need to remain in wireless connectivity range to store this data. This adds another device to our solution, so we do not consider it.

To store the roughly 6 Megabytes of data, we need some kind of memory. We see in 2.1.7 that our microcontroller only has a Flash Memory of 48KB. This would mean we need a memory to store the data during the interval that the user is wearing the device. This data can then be transferred to a computer from the memory once we have a connection between the two.

We compared the available options in the market. Most memory chip's are sold in power's

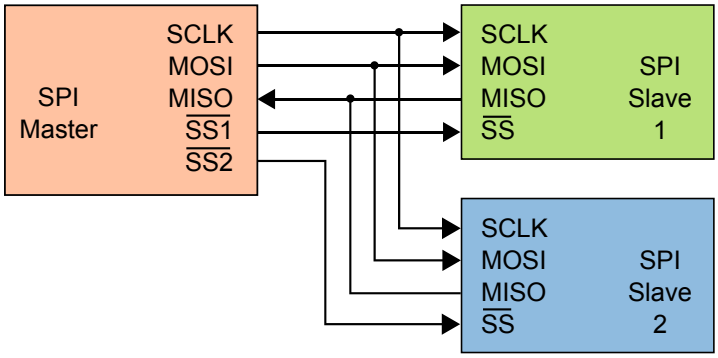


Figure 2.6: An SPI system with two slaves on the same bus.

of 2, so we would have to look at devices close to 8 Megabytes or 64 Megabits in design. A search on Digikey in October 2013 showed that there were very limited options in the Flash memory market that would support such a high density of memory. The options were further limited by the fact that we required a small footprint that we could add to our wrist mountable device. Options typically had parallel input / output which meant they had close to 28 pins.

Atmel's memory division had just released AT45DB642D, a 64-megabit 2.7V Dual-interface DataFlash memory chip. This chip would support the amount of memory we required, and ran on a single 2.7V - 3.6V supply, which our system worked on. This chip also supported SPI, the same protocol that was being used by the sensors in our circuit. SPI allows us to use the same bus for data communication and clock, while using one extra line for each device's slave select pin. Figure 2.6, courtesy of [11] shows how two slave devices can connect to an SPI master. This means that we could reduce the number of I/O pins used in our microcontroller, and also reduce the number of tracks created on our PCB for data communication.

The chip was made available in an unconventional CASON package, which is a surface mount device. This means we would have to solder the chip before it could communicate with a microcontroller. To avoid the delay incurred because of PCB fabrication (which can take up to two weeks), we jerry rigged the chip. After flipping it upside down, so that its pins are exposed, we soldered strands of a multistrand wire to each of these pads. These strands were then soldered through a dot matrix PCB, allowing us to use the new unit as a Pin Through Hole device. This required careful handling of the chip with a dextrous hand, and was done under a microscope. Figure 2.7 shows the result.

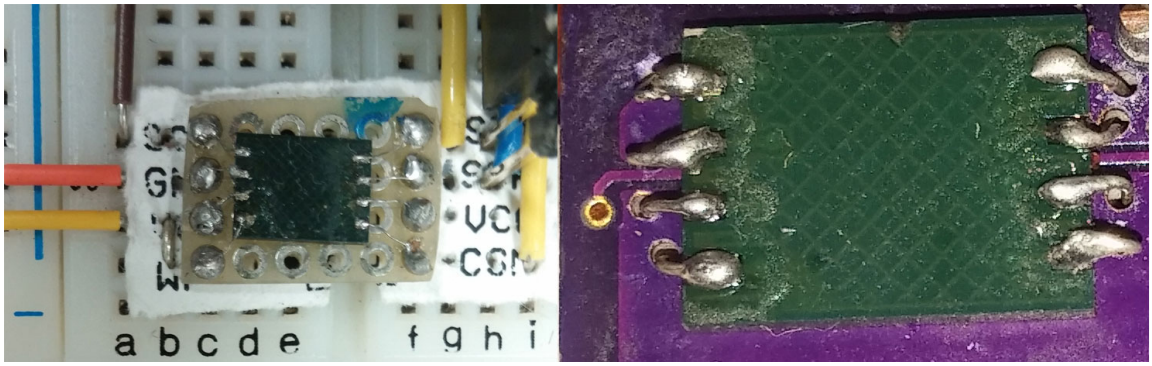


Figure 2.7: Two examples of flipping over a CASON chip and soldering wires to it's pads.

2.1.7 Microcontroller

The sensors we discussed in the previous section cannot function on their own. They are primarily slave devices that can be requested for data. To get this data, we require a master device, which we have in the form of a microcontroller. Our microcontroller would need to support the communication protocol used by our sensors and memory chip. It would also need to have a sufficient memory to be able to temporarily buffer sensor data before it is flushed to the external memory chip.

Several manufacturers provide microcontrollers that we can use, however, our priority of low current is what will separate the different microcontrollers, and help us pick the right one for our device. We considered the different wrist based devices on the market and realized that the MSP430 family of microcontrollers greatly dominates this market segment. The MSP430 Brochure [12] lists roughly three hundred parts that can be picked from, categorized as:

- Low Power FRAM Series
- Value Line Series with limited components
- F - Family with increased integration and performance
- F - Family with an integrated LCD driver
- F - Family with high frequency operation (up to 25 Mhz)
- CC430 Series (Microcontroller with integrated RF chip)

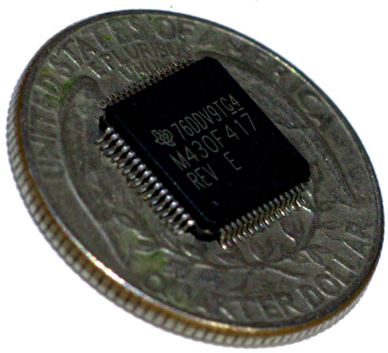


Figure 2.8: The MSP430F248 integrated chip, with a quarter to show scale.

When selecting our chip, we needed one that would allow future expansion with regards to the capabilities of the device. It would also need to have a buffer large enough to store a sizable amount of data before it would need to be flushed to the memory chip. After combing through the brochure, we found the first parts with SPI support and a minimum SRAM of 2048 bytes in the MSP430F24X series. This sub-family of devices had the following characteristics (as seen in [13])

- Low Supply-Voltage Range, 1.8 V to 3.6 V
- Active Mode Current Consumption: 270 A at 1 MHz, 2.2 V
- Standby Mode (VLO) Current Consumption: 0.3 A
- Four Universal Serial Communication Interfaces supporting SPI and I^2C.
- Two 16-Bit Timers with multiple compare registers.

We select the MSP430F248 as our microcontroller. This chip comes with a 48KB+256B Flash Memory, and 4KB of RAM. Assuming our program size would not be more than 1 KB, this would allow us to buffer about 3KB of data from the sensors before flushing it to the external memory, allowing the memory to remain in sleep mode for a large amount of time.

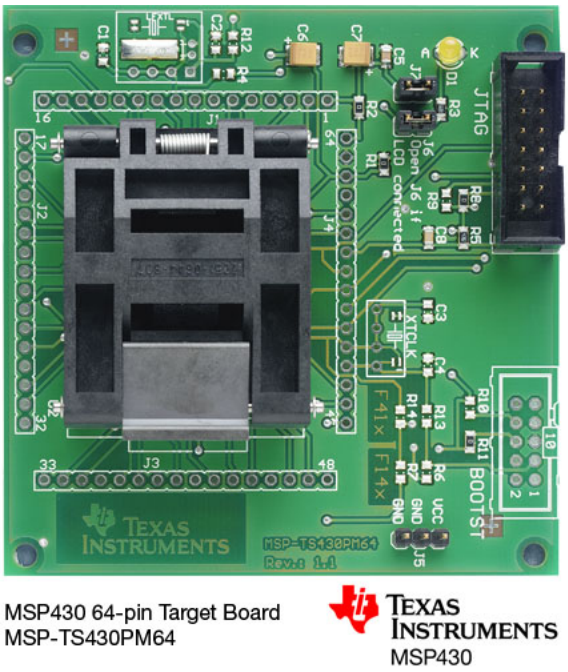


Figure 2.9: The MSP430 64-Pin Target Board. Image provided by Texas Instruments

2.1.8 MSP 430 Target Board

Like our sensors, the MSP430F248 also comes in a surface mount package. The footprint is shown in Figure 2.8 and known as the QFP[3] design. This package does have pins, but they are not long enough to pass through a PCB. Instead they have to be mounted on the PCB. It is very hard to design a complete system without bugs and if we would design a PCB and write microcontroller code at the same time to operate the sensors and the memory, it would not be easy to debug an error because we would not know if the error is in the hardware design or the software code. Texas Instruments' understands this, and provides a 64-Pin Target board for its microcontroller. This target board has a ZIF[4] socket that accepts the 64-Pin QFP chip, and breaks out its pins. This means other devices can be connected to the board with ease, allowing us to program the microcontroller without having soldered it.

Figure 2.9 shows us what the MSP430 Target Board looks like, with the socket in the center. You can see a connector in the top-right of the figure, which is labeled JTAG. The socket is surrounded by holes on all sides which connect to the microcontrollers pins, allowing for easy

[3]Quad Flat Package
[4]Zero Insertion Force

Figure 2.10: The MSP430 Flash Emulation tool. Picture courtesy Texas Instruments

connections. The target board also allows for a crystal to be soldered next to the microcontroller if the programmer wants to use the Low Frequency Clock based on an external crystal. We use this crystal for and accurate clock when polling our sensors, and also when communicating with a host computer as we will see in section 2.1.10

2.1.9 MSP430 Flash Emulation Tool

Now that we have a microcontroller selected, and a target board that allows us to easily set it up electrically, we need a way to program it with our code. The MSP430 series of microcontrollers are programmed by using JTAG using 4 wires. Modern computers usually lack a serial or parallel port for communication, so Texas Instruments provides programmers with the MSP430 Flash Emulation Tool. This tool (seen in Figure 2.11) connects to a desktop computer through USB, and the other end connects to a standardized JTAG connector, as seen in 2.9.

The tool also supports powering the system it is programming, so we were able to program our microcontroller without an external power source. Once we had our battery selected, a jumper on the MSP430 Target Board allowed us to disable the power supply of the Flash Emulation Tool. The FET can be used by various software, paid and free. For our use, we picked IAR's Code Composer Studio as it was available free of cost to students. Through the MSP430FET, IAR supports not only programming and flashing the microcontroller, but also debugging the code while it is running. This feature enabled us to fix multiple bugs in a short amount of time, something that would be hard to do without the debugging capabilities. We can also inspect different sections of the memory, allowing us to see live how the code is behaving.

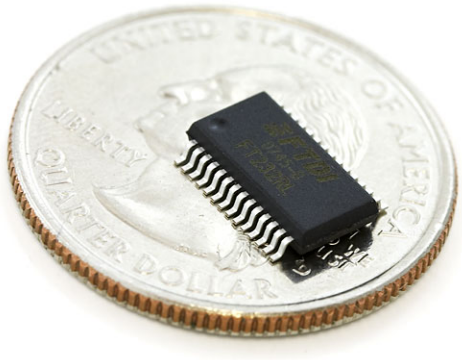


Figure 2.11: The FT232RL USB to UART bridge. Picture courtesy Sparkfun.com

2.1.10 USB to UART Bridge

With the above hardware in place, hopefully we have a system which reads from sensors and stores this data in the memory chip. Once data for an entire day is logged, we would like to transfer the contents of this memory to a computer where the sensor data can be analyzed for different patterns and behvaiours. Broadly speaking, there are two methods to transfer data, wireless and wired. Both these methods required parts that consume a current of about 18 mA when actively in use. Wireless transfer of data would consume power from the battery (unless a wire is connected to power the system), and assuming a speed of 115Kbps, this would take about 15 minutes to transfer. With an active current draw of 18mA for most wireless technologies, this would require a huge amount of power, something that the battery cannot sustain.

We solve both these problems by using a wired connection to transfer data. As mentioned in 2.1.9, modern computer systems only have the USB port as a viable way to communicate. Although some MSP430 microcontrollers have native USB support, those microcontrollers have an increased current consumption and are not viable to our purpose. Our selected microcontroller, the MSP430F248, does not have a native USB module, but does have other communication modules. The MSP430F248 has a UART, and a combined I^2C + SPI module that can be used for communication. Since we will be using SPI for communicating with the sensors and the memory chip, using

UART is a good way to isolate modules being used at the same time.

We looked for devices that would help convert communication in one protocol to another. Since SPI, I^2C and UART are all serial communication protocols, in theory, any device that can communicate in one of these protocols should be able to communicate in the other. However, a search on different devices in the market showed that USB to UART bridges are an easy way for microcontrollers to communicate with a computer. The device is detected as a virtual serial communication port to the operating system, and can be used as a regular serial port. Searching for USB to UART devices shows two primary contenders in the market segment, Silicon Labs and Future Technology Devices International (also known as FTDI). Both offer devices that have similar features and specifications.

We need a device that would communicate easily with our microcontroller, the MSP430F248 and also be easy to prototype with. All of Silicon Labs' offered parts are only available in the QFN package, which means it would be hard to prototype using them. For this reason, we select FTDI's chip, the FT232RL. FT232RL has the following specifications (as seen in [14]):

- Single chip USB to asynchronous serial data transfer interface.
- Data transfer rates from 300 baud to 3 Mbaud (RS422, RS485, RS232) at TTL levels.
- UART interface support for 7 or 8 data bits, 1 or 2 stop bits and odd / even / mark / space / no parity.
- Integrated +3.3V level converter for USB I/O.
- Available in compact Lead-free 28 Pin SSOP and QFN-32 packages.

2.1.11 Battery

While the wrist activity monitor is mounted and active, it needs a power source to operate on. Work done by [15] uses a single 450mah Lithium Ion battery. These batteries are

Battery Type	NiCd	NiMH	Lead Acid	Li-ion	Li-ion polymer	Reusable Alkaline
Cycle Life (to 80% of initial capacity)	1500	300 to 500	200 to 300	500 to 1000	300 to 500	50
Fast Charge Time	1h typical	2-4h	8-16h	2-4h	2-4h	2-3h
Overcharge Tolerance	moderate	low	high	very low	low	moderate
Self-discharge/Month (room temperature)	20%	30%	5%	10%	10%	0.30%
Cell Voltage(nominal)	1.25V	1.25V	2V	3.6V	3.6V	1.5V
Peak Load Current	20C	5C	5C	>2C	>2C	0.5C
Preferred Load Current	1C	0.5C or lower	0.2C	1C or lower	1C or lower	0.2C or lower
Maintenance Requirement	30 to 60 days	60 to 90 days	3 to 6 months	not req.	not req.	not req.

Table 2.2: Comparison of Various movement sensors available in the market.

Table 2.2 shows the different current battery types available (as seen in [16]), and their parameters.

2.1.12 Soldering

Appendices

Bibliography

[1] Y. Dong, J. Scisco, M. Wilson, E. Muth, and A. Hoover, "Detecting periods of eating during free living by tracking wrist motion," 2013.

[2] Y. Dong, A. Hoover, J. Scisco, and E. Muth, "A new method for measuring meal intake in humans via automated wrist motion tracking," *Applied psychophysiology and biofeedback*, vol. 37, no. 3, pp. 205–215, 2012.

[3] C. L. Ogden and M. D. Carroll, "Prevalence of overweight, obesity, and extreme obesity among adults: United states, trends 1960–1962 through 2007–2008," *National Center for Health Statistics*, vol. 6, pp. 1–6, 2010.

[4] M. J. Mahoney, "The obese eating style: Bites, beliefs, and behavior modification," *Addictive Behaviors*, vol. 1, no. 1, pp. 47–53, 1975.

[5] S. A. Bowman, S. L. Gortmaker, C. B. Ebbeling, M. A. Pereira, and D. S. Ludwig, "Effects of fast-food consumption on energy intake and diet quality among children in a national household survey," *Pediatrics*, vol. 113, no. 1, pp. 112–118, 2004.

[6] A. M. Andrade, G. W. Greene, and K. J. Melanson, "Eating slowly led to decreases in energy intake within meals in healthy women," *Journal of the American Dietetic Association*, vol. 108, no. 7, pp. 1186–1191, Jul 2008, id: 1; JID: 7503061; 2007/04/24.

[7] M. Drennan, "An assessment of linear wrist motion during the taking of a bite of food," 2010.

[8] Sparkfun, "Sparkfun Tutorials: How a Gyroscope Works," Last Accessed: October 2014.

[9] ——, "3-Axis Gyro/Accelerometer IC - MPU-6050," Last Accessed: October 2014.

[10] ——, "Triple Axis Accelerometer and Gyro Breakout - MPU-6050," Last Accessed: October 2014.

[11] C. Burnett, "SPI Three Slaves," Last Accessed: October 2014.

[12] T. Instruments, "Msp430 ultra-low-power microcontrollers," *MSP Product Brochure*, 2009.

[13] ——, "Mixed signal microcontroller," *MSP430F248 — MSP430F2x/4x — Ultra-low Power — Description and parametrics*, 2009.

[14] F. T. D. I. Ltd, "Ft232r usb uart ic," *FT232R USB UART IC*, Last Accessed: October 2014.

[15] S. Research, "Shimmer Sensing," Last Accessed: October 2014.

[16] ——, "Shimmer Sensing," Last Accessed: October 2014.