

# Love Watch 1702

ELEC 327 Final Project

Yoseph Maguire, Yovahn Hoole, Chandler Burke

## Abstract

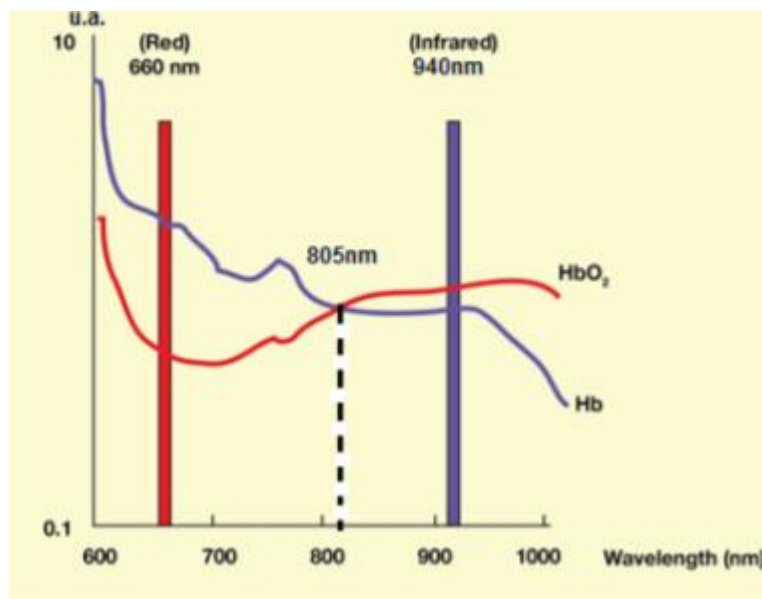
Love Watch 1702 is a smart watch capable of measuring heart rate, oxygen saturation and most importantly the presence of loovvee. We three members of the Lonely Hearts Club, Yoseph Maguire, Yovahn Hoole and Chandler Burke have come up with a unique and state of the art system to help a user detect if they are in love by detecting a spike in heart rate. This research has far reaching consequences in protecting young ELEC's from the distractions of college life.

## Pulse Oximeter

Oxygen is passed throughout the body by means of the hemoglobin in our blood. The hemoglobin within our blood can either be oxygenated (carrying Oxygen) or deoxygenated (not carrying Oxygen). The concentration of oxygenated vs. deoxygenated hemoglobin is the basis for pulse oximetry.

### Physical properties of Blood

Pulse oximeters utilize the reflection and absorption of light to measure the oxygen saturation of a user's blood. Oxygenated hemoglobin is distinctively red and deoxygenated hemoglobin is a distinctly dark blue. The optical properties of blood are highly dependent on the amount of oxygen within the body. When blood is exposed to a light source, the oxygenated hemoglobin content determines the amount of light that is absorbed and reflected back. As can be seen in *Figure 1* the absorption of red light dips when exposed to oxygenated hemoglobin while the absorption of Infrared light drops when exposed to deoxygenated hemoglobin.



*Figure 1: Oxygenated vs. de-oxygenated blood light absorption of IR and Red*

## Measurements

The pulse oximeter uses an LED and a sensitive photodiode to measure these differences in reflection of red and Infrared wavelengths to measure the oxygen content of blood. The photodiode converts the reflected light into an electric signal that is proportional to the intensity of the light and an ADC converts this into a digital signal.

## Heart Rate Measurements

To measure heart rate, the pulsatile nature of blood can be used. In systole, when the heart muscle contracts and pulsation is at its peak, the volume of blood in the tissue increases. The increased volume of blood leads to more light being absorbed. In diastole, less blood is present in the tissue. The reduced volume of blood allows for less light to be absorbed and more to be reflected back to the pulse oximeter. The MAX30105 uses an IR sensor to collect relative reflectance information which can be processed to obtain a simple heart rate metric.

## Hardware

The Love Watch 1702 utilizes MAX30105 particle sensor and an Adafruit 14-Segment LED Backpack driven by a HT16K33 chip.

### Adafruit LED Backpack

In order to output the estimated heart rate, we obtained a 4-digit 14-segment display that could accept I2C communication. We decided to use the 0.54" 14-segment LED HT16K33 Backpack from Adafruit, as this can be used with I2C to create alphanumeric letters.

The controller can be controlled via an Arduino library, making it easy to interface with and update when necessary. The code allows for control over each segment of each block in the Backpack, making it useful for personalized messaging. Also, the LED's are very bright and adjustable. In short, this device is a good choice as it allows one to very quickly and efficiently output numbers using the I2C protocol.

### MAX30105 Particle Sensor

The MAX30105 particle sensor include internal LEDs, photodetectors and ambient light rejection. The product is quite small and can easily be used for wearable devices such as the Love Watch 1702, fitting right under the wrist of the user. Furthermore the presence of ambient light rejection is an important feature for Love Watch since external light sources can easily interfere with the readings of the sensitive photodiodes needed to measure oxygenation. To preserve long battery life, the MAX30105 also allows for Ultra-Low Power operation.



Figure 2: MAX30105 Particle Sensor

## I2C Communication

Both the MAX30105 particle sensor and the Adafruit LED Backpack utilize I2C communication. The main advantage I2C requires only two pins and allows for multiple slaves off of those pins. I2C allows for multiple slaves on the same bus by addressing.

## Protocol

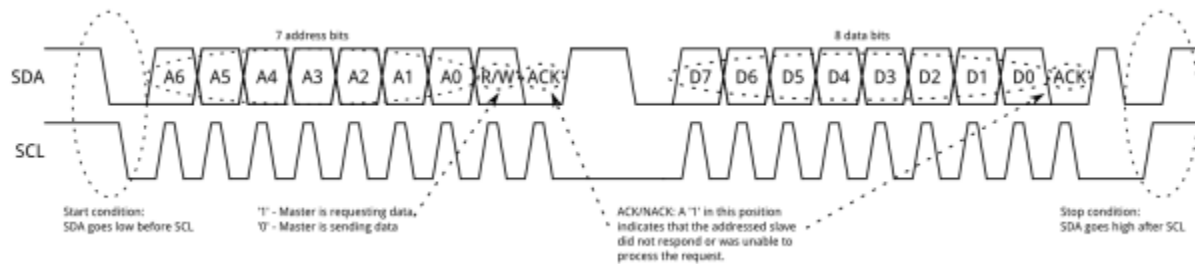


Figure 3: Protocol for I2C communications – Sparkfun I2C tutorial

Messages in I2C are broken up into two types of frames: an address frame and one or more data frames which are 8 bits each. In the address frame, the master indicates the address of the slave which should be receiving the message and in the data frames the master sends the actual message. To initiate the address frame, the master sends a start condition which notifies all slaves on the bus that a transmission is about to start. Following the address frame, master specifies an R/W bit indicating whether this following operation is a read or a write. Between each frame the recipient of the message sends an Acknowledge by the Receiver message. Once the Data Frames are sent the master also issues a stop condition which notifies the slave that the message has been sent and relinquishes control of the bus.

## MAX30105 Protocol

To write to the MAX30105. Following the I2C addressing requirements the master must send 2 bytes via. I2C protocol including first the address of the register to be written to and second the value to be written. These would each be sent in a separate data frame. This process can be seen in Figure 4 from the MAX30105 datasheet.

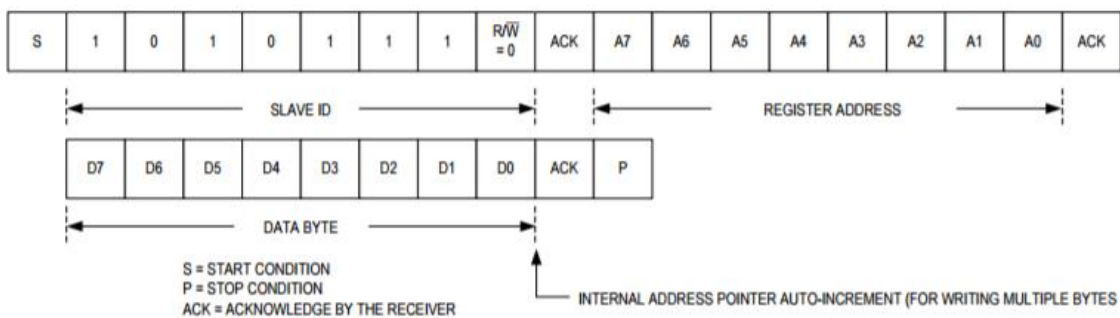


Figure 4: Writing One Data Byte to the MAX30105

Reading on MAX3015 is slightly more completed. The master must first send the register address to be read followed by a repeated start condition. From here the Slave will respond with its Slave Address followed by the data to be read in multiple data frames. Once the master has read the information in requested, it must further send a Not acknowledge followed by a stop condition. The process can be seen in Figure 5 from the MAX30105 datasheet.

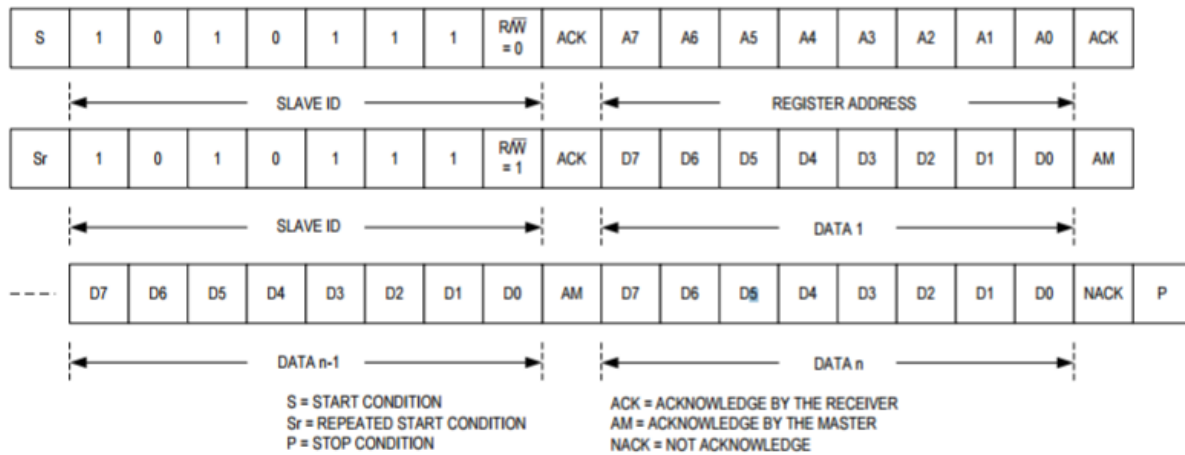


Figure 5: Reading multiple bytes of data from the MAX30105

### Adafruit Backpack: HT16K33 Protocol

The Adafruit backpack uses a fairly simple I2C Protocol. Following the I2C addressing requirements the master must simply send the desired command code to the backpack.

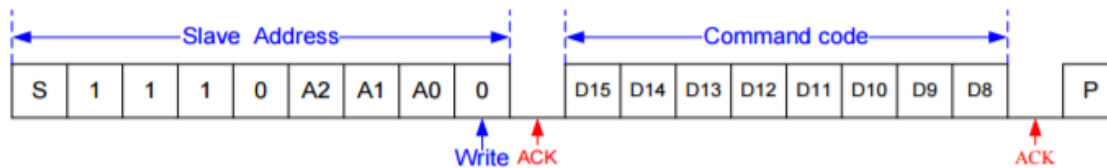


Figure 6: HT16K33 Protocol - HT16K33 datasheet

### Design Challenges

During this project our team encountered significant design challenges. First, we originally designed our PCB for a 128x64 OLED Display. When programming the OLED display, we realized that the buffer size necessary to render images would be pushing the capabilities of the MSP430. The libraries used to generate graphics for the OLED display also were written for Arduino, so we would have to create our own libraries for all parts of the display. Because the display was not the core functionality of our project, we decided to pivot and opt for a 14-Segment Display instead, which would be more manageable to program. The decision to change displays also meant that we could no longer use our PCB, since the wiring on the two displays is vastly different.

Beyond the display, we struggled with writing an I2C library to use with our sensor and display. In order to perform signal

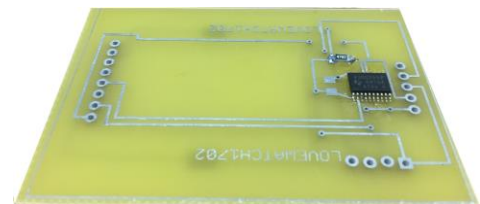


Figure 7: PCB. Unused in final product.

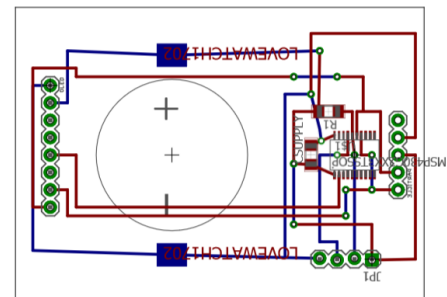


Figure 8: Original PCB Design

processing on the photodiode data stream, we needed to write to the MAX30105 and then read the values. To read from the proper registers required a Start – Restart I2C format. Additionally, the LED Backpack required a simple Start format for I2C communication and did not need to be read from. Unfortunately, we were unable to get past our design of I2C, and were left to abandon a buggy design. Our final product consists of an Arduino instead of an MSP430 for ease of programming, and a breadboard circuit since we were unable to order a new PCB in time.

Through this project, we learned about the significant challenges associated with every part of a design. Things we took for granted, like graphics libraries for displays, proved to be buggy or nonexistent, and programming our board from the ground up forced us to gain confidence in every aspect of the microcontroller. Debugging especially forced our group to work through datasheets and determine line by line the purpose of our code and what the MSP430 required for I2C communication.

## Final Product

After pivoting drastically due to the failure of the I2C MSP430 Library we wrote, among other things, we ended up with a design based on an Arduino. Represented in breadboard version, the system uses I2C to read infrared values from the MAX30105 sensor, and then after using those values to calculate heart rate, send back graphical representations of heart rate to the 14-Segment Display. To model a simple threshold algorithm, if the user heart rate exceeds 80 BPM, it triggers a “Love” animation, indicating to the user that they have the potential to be in love (note, this system has an extremely high false positive rate).

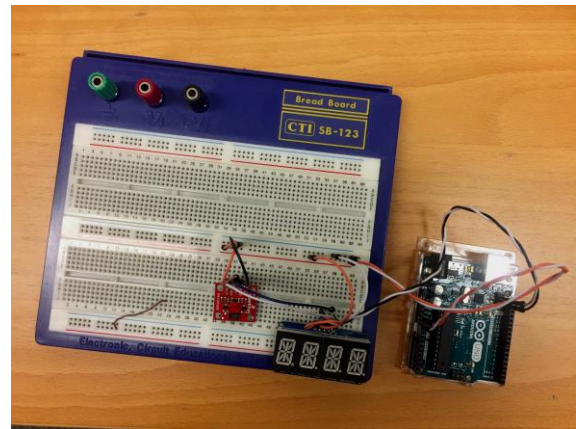


Figure 9: Final Design with Arduino

## References

Adafruit. “Adafruit/Adafruit\_LED\_Backpack.” GitHub. N.p., 05 Dec. 2016. Web. 04 May 2017.  
“MAX30105\_Particle\_Sensor\_Breakout.” N.p., n.d. Web.

“Miniaturized Pulse Oximeter Reference Design.” TIDA-00311 Miniaturized Pulse Oximeter Reference Design (n.d.): n. pag. Www.ti.com. Web.

Ram Mapping 16\*8 Led Controller Driver With Keyscan. “RAM Mapping 16\*8 LED Controller Driver with Keyscan.” HT16K33 (n.d.): n. pag. Adafruit. Web.

SFUPTOWNMAKER. “I2C.” I2C. N.p., n.d. Web. 04 May 2017.

Sparkfun. “Sparkfun/MAX30105\_Particle\_Sensor\_Breakout.” GitHub. N.p., 01 Dec. 2016. Web. 04 May 2017.

Tilakaratna, Prasana. “How Pulse Oximeters Work Explained Simply.” HowEquipmentWorks.com. N.p., n.d. Web. 04 May 2017.