Metrigloves: The Digitization of Measurement Instruments within the Construction Industry

Jasper K. Lam *(CSE)*, Raymond Yu *(CpE)*, Gabriel Quach *(CSE)*
Advisor: Professor Stuart Kleinfelder

***Abstract*— Since 2007, the Arduino platform has provided an easy to use interface that combines high level hardware integration with relatively simple coding logic. With this in mind, we implemented and incorporated the versatility of Arduino technology into common, everyday construction gloves. By using digital sensors, we have replaced the use and need of physical measurement instruments such as the tape measure, thus reducing the overall strain or hassle on a project for construction workers.**

After extensive testing and research, the finished glove comes with 3 fully implemented sensors: a gyroscopic, distance, and RGB value color sensor. The gyroscopic sensor replaces the need for a leveling tool and allows the user to detect if something is perfectly flat or level, and if it is not, show how much it is off. The distance sensor replaces the need for a tape measure by detecting the distance between the sensor and the surface in front of it. The color sensor returns the RGB values that make up the detected color, allowing for the wearer to replicate that exact color when shopping in a store or doing research. All of the values and readings are then displayed on an LCD screen on the back of the hand, and the screen allows for switching between each mode using the LCD menu.

***Index Terms*—Arduino, Metrigloves**

I. Introduction

Inaccurate measurements paired with improper equipment/preparation in construction can be fatal. Construction worker deaths on site represented 21% of all private industry deaths with 39% of construction deaths being falls [1]. With the construction site already being a common hazardous environment, highrise projects accentuate the risk of death greatly. Furthermore, a conducted study by the Center for Construction and Research found that back pain constitutes 16.4% of injuries [2], which exposes an issue with the tool belt. A fully loaded tool belt can weigh up to 50 lbs with some of those tools being measuring instruments. Twisting to reach a tool from the belt with a 50 lb load can strain the body. Doing this while working at a height can prove to be fatal.

The purpose of the project, Metrigloves, was to discover a way to make it easier and safer for construction workers to get their job done by offering them a variety of tools for their hand without compromising the gloves main function: protect the hand. Attaching lightweight measuring sensors on a pair of construction gloves will reduce the burden on the tool belt, thus reducing the risk of falls, back, and other types of injuries. Metrigloves combines construction tools with modern day technology to make measurement and other simple recognition tasks seamless, thus improving their efficiency and quality of work.

While there have been Arduino-glove projects found on the Internet, there aren't any that attempt anything similar to this project's goals of integrating so many different aspects into one. However, of course, there are many Arduino projects that involve the components that we used for our gloves (distance sensing, color reading, gyroscopic measurements, etc.). Therefore, these previous guides and tutorials were used as a frame of reference and learned from. Our methods included studying from these tutorials to implement our hardware (wiring) and software (code) and making the appropriate changes as we saw fit. We also decided to choose small, lightweight components to prevent burdening and overall inconvenience.

After testing and reevaluating design choices, the original decision to split the sensors between two gloves was changed since we discovered a way to get everything comfortably on one glove. The finished product features an LCD screen on the back of the hand to display results, and that same LCD screen also handles the user interactions to switch between each detection mode. The distance and color sensor are shared across the knuckles of the glove, while the gyroscopic sensor is near the LCD. Over the course of two quarters, we were able to get appropriate and accurate readings for all three types of sensors, and make the successful integration of hardware and software onto the glove. The original intention of having a screw detector using a camera was also removed as that is something that a phone could do, which is also easily accessible and very common for anyone working to have, thus being redundant and unneeded for the glove. Furthermore, components were limited to at most three sensors paired with an LCD due to limited pins on an Arduino and limited spacing on a breadboard.

II. Metrigloves

Metrigloves is a multipurpose multi-measuring glove designed for the convenience of construction or home project workers. The glove is equipped with a wide range of tools while still maintaining the ability to lift and grasp objects. We have decided to make our project purely Arduino-based due to the wide range of devices that are compatible with Arduino, and also we want to take simple, programmable devices such as an Ultrasonic sensor and build something more complex.

III. Materials

1) *Arduino Nano v3.0*

   Smaller Arduino circuit board programmable through the Arduino IDE. The Nano was chosen over the Arduino Uno due to a substantial size difference.

2) *Adafruit TCS34725 (Color Sensor)*

   Reads the color value (hexadecimal or RGB array) of a surface. Particularly useful for matching/choosing a desired color for wall painting.

3) *HC-SR04 Ultrasonic (Distance Sensor)*

   Measures the distance between the sensor and the surface in front of it with a max range of 4 meters. Very useful tool if quick measurements (without measuring tape or ruler) are needed. Ultrasonic is chosen due to its reliability, accuracy, and ease of use.

4) *IMU Breakout - MPU 6050 (Gyroscope)*

   Measures orientation (x, y, z position) using 3 different axes and also measures acceleration.

5) *2.0" 320x240 Color IPS TFT Display (LCD)*

   Variation of an Adafruit ILI9341 LCD that we used to display the measurements of the sensors. Chosen for its various pixel-based drawing features and touch input sensing from its ILI9341 library.

6) *Handyman Flex Grip Work Gloves*

   Construction gloves used for our project.

IV. Hardware

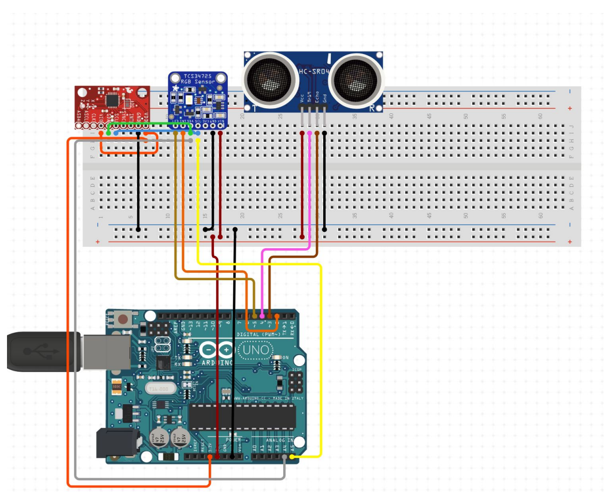

Fig. 1. General wiring diagram of the components used for the left glove

All the components are connected to the 5V power bus. The wiring allows us to ensure each component receives 5V without any voltage drops. Each of the parts occupy a port on the Analog or Digital Pwm since analog pins can be used digitally. SDA and SCL of both the color sensor and gyroscope must be connected to A4 and A5, respectively, based on the I^2C standard. All these configurations allow us to read each of the sensors separately to guarantee accuracy and consistency. It also makes each of the components independent of each other so in case any of the modules malfunction. It will not affect the performance of the rest of the system; however, of course it is necessary that we use a common voltage source node and common ground node. Therefore, connections to VCC and GND must be ensured before usage.

Wiring Port Tables

- Blank entries are unused ports.
- The Bus POS (5V) and GND are wired to VIN and GND pins, respectively, on the Arduino Nano.
- D/A - can be connected to any digital or analog pin.

A. HC-SR04 Ultrasonic

VCC	Trig	Echo	GND
Bus POS (5V)	D/A	D/A	Bus GND

B. Color IPS TFT Display

VCC	GND	CS	RST	DC
Bus POS (5V)	Bus GND	D/A with 10k resistor	D/A with 10k resistor	D/A with 10k resistor
MOSI	**SCK**	**LED**	**MISO**	**T_SCK**
D/A with 10k resistor	D/A with 10k resistor	D/A	D/A	D/A with 10k resistor
T_CS	**T_DIN**	**T_DO**	**T_IRQ**	
D/A with 10k resistor	D/A with 10k resistor	D/A	D/A with 10k resistor	

C. TCS34725 RGB Color Sensor

LED	INT	SDA	SCL	Vin	GND
		Analog In A4	Analog In A5	Bus POS (5V)	Bus GND

D. Gyroscope Sensor

SDA	**SCL**	**Vin**	**GND**
Analog In A4	Analog In A5	Bus POS (5V)	Bus GND
XDA	**XCL**	**ADD**	**INT**

V. Software

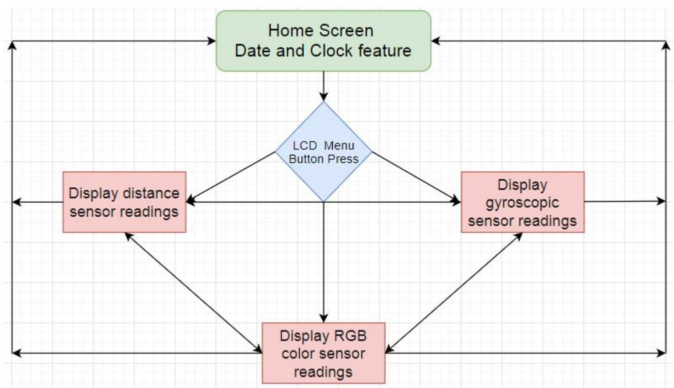

Fig. 2. Software flowchart that provides the general architecture of our Arduino program.

The software architecture featured which allows the user to control these sensors consists of four states. When powering on the gloves, the user is greeted with a home screen displaying the current time date. On the homescreen, there are 3 buttons each labeled with the sensor name. Once the corresponding sensor has been selected, it will start to read values. A secondary tap of the display will return the user to the home screen. or to a different sensor if the user presses that specific sensor's button. To preserve battery, the display will automatically return to the home screen after 2 minutes.

VI. Methods

A. Implementation and Testing

Our approach to implementing code for the sensors involved learning from videos/tutorials to understand how to connect and program the sensors, and then from there we would adapt the program to fit our design. HowToMechatronics.com provides written guides, wiring diagrams, Youtube tutorials, and example code that assisted us in this process [4]. After finishing each stage of software and hardware implementation, we tested each sensor individually for accuracy and consistency. When the sensors were all finished, testing and creation of the LCD home display and changeability of modes was the next step that was completed. Once all of the hardware and software for the LCD and sensors were finished, integration of hardware onto the glove was planned, then tested, to make sure everything would still work and be wired properly when on a glove.

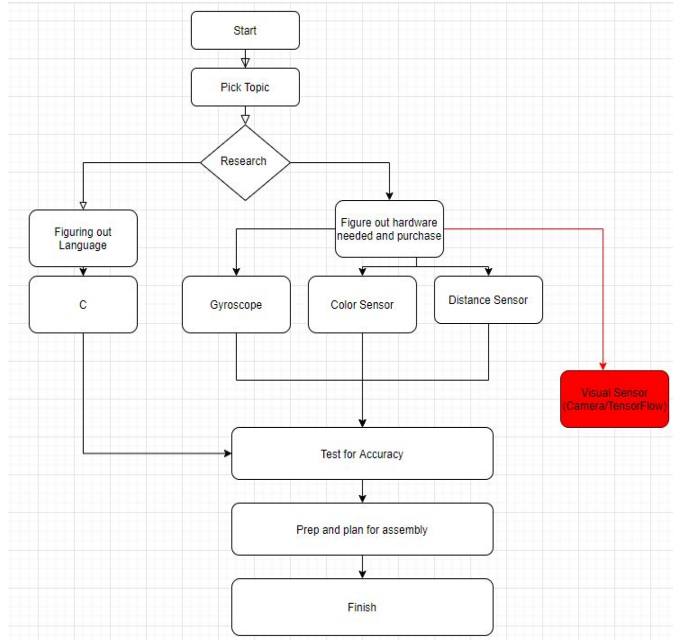

Fig 3. The overall planning and project path over the course of two quarters

B. Arrangement

Fig. 4. Simplified image of the arrangement of components on the glove.

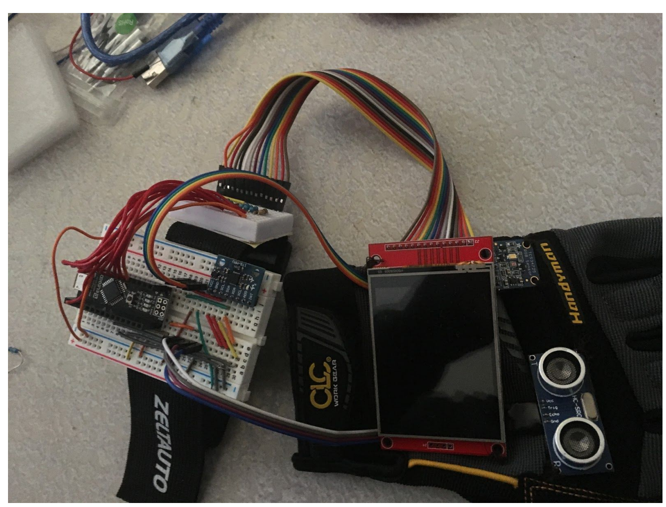

Fig. 5. Image of the glove.

Fig. 4 shows the general arrangement of our hardware. All components fit on the back of the hand to allow for grasping and unimpaired function of the actual glove. The LCD is on the back of the hand, while the ultrasonic and color sensor reside on the knuckles of the glove, above the LCD. The rest of the wiring goes on the breadboards which ends up being attached below the glove on a separate velcro wristband.

VII. Results and Performance

A. HC-SR04 Ultrasonic

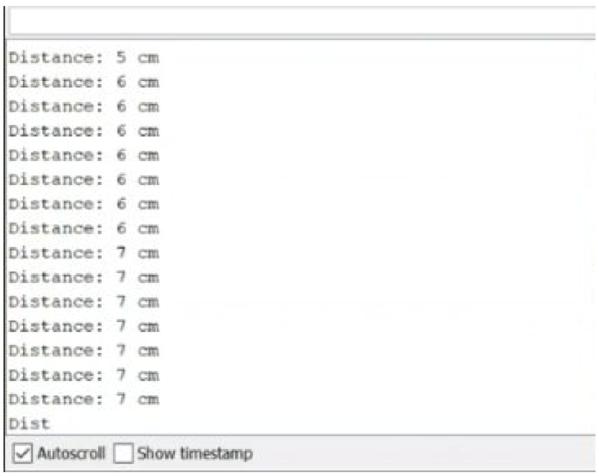

Fig. 6. Measurements taken by the Ultrasonic outputted to the serial monitor. Shows a consistent measurement of 6-7 cm.

Fig. 7. Testing the distance sensing capabilities of the Ultrasonic using a printable ruler as reference. The ruler's bottom scale is centimeters.

The formula for calculating distance is

$$distance = duration * 0.034 / 2$$

where we multiply the travel time of the pulse wave (sent by the Ultrasonic Trig and Echo) by the speed of sound (in $cm/\mu s$) divided by 2 (since the wave travels to the surface and propagates back).

According to the testing shown by Fig. 6 and 7, we can see that the Ultrasonic has a +/- 1 cm margin of error at close ranges.

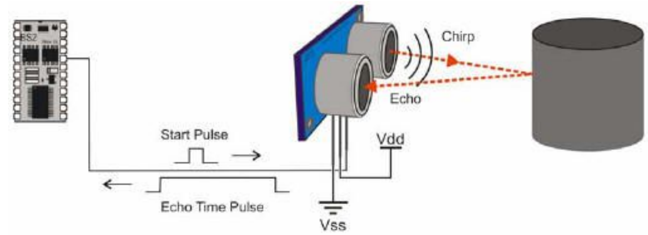

Fig. 8. Shows how the Ultrasonic used the Doppler effect to measure distance.

Using concepts of physics, we can assume this margin of error is due to outliers such as other sound waves interfering with the initial wave. Another factor the ultrasonic distance sensor doesn't account for is the surface of the objects as it can change the trajectory of the echo coming back. We were able to draw these conclusions using physics because when calculating the distance on paper, we ignore these factors or deem them as negligible. The margin of error does increase the further away the surface is but we can safely say that the accuracy is above 90 percent. When measuring distances of 1-4 m without a tape measure, this data is considered reliable.

Fig. 9 . Distance measurements displayed on the LCD that ends up being attached to the glove.

B. Adafruit TCS34725

Fig. 10. Portable charger used for testing color sensor accuracy

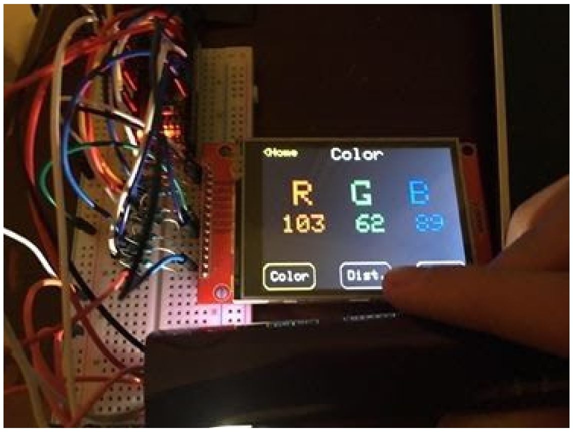

Fig. 11. RGB values of the portable charger as sensed by the color sensor

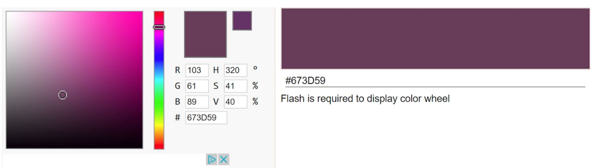

Fig. 12. RGB color display from *colorspire.com*. Color values displayed by Fig. 11 are inputted to show the color sensed by the TCS34725.

According to Fig. 10, 11, and 12, the color of a purple portable charger was sensed by the color sensor and it was shown that the color sensed does resemble that of the charger. We can say that the TCS34725 senses quite accurately under the right conditions, given proper distance away and lighting for the sensor to pick up the object under detection.

Although the color sensor returns RGB values that resemble the color of the object, the human eye can notice a difference between the measured color and the actual one. This slight variation in color is due to unideal lighting conditions which can influence the measured value. We can conclude that in order to measure color with consistently accurate results, we would need to have a controlled environment where we can limit the amount of influencing factors. This is a potential undersight of the project, and a possible fix is shining an extra light from the phone to help with the best possible lighting for best results of color detection, or to simply upgrade the hardware with a higher quality product.

Fig. 13. Color RGB values displayed on the LCD

C. MPU 6050 Gyroscope

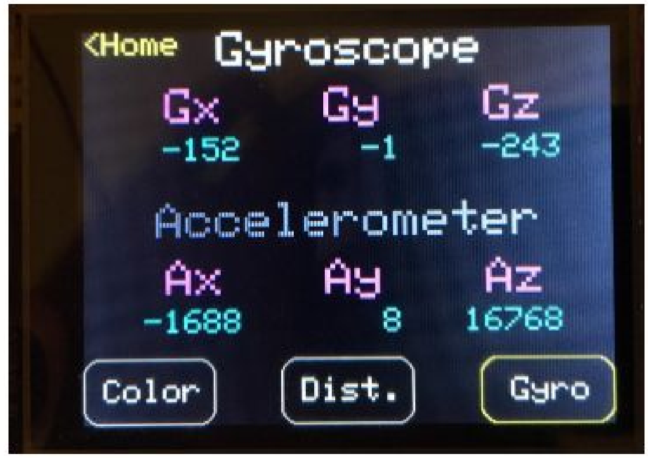

Fig. 14. The readings on the LCD from a sample of the gyroscope and accelerometer

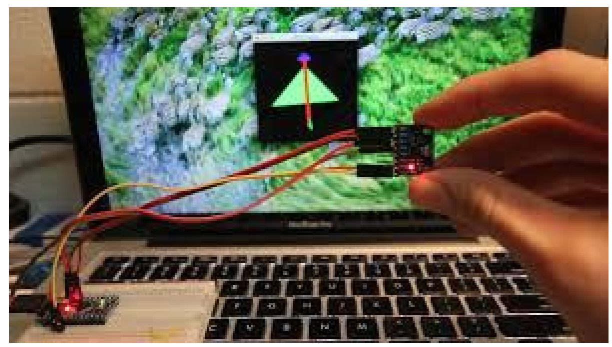

Fig. 15. MPU 6050 Teapot Demo shown on PC using Processing 3 and Gyroscope Readings

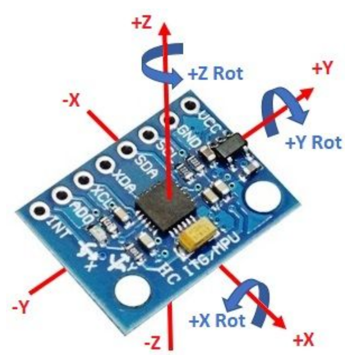

Fig. 16. Diagram of the gyroscope (x, y, z) axes to better understand readings

According to Fig. 14, 15, and 16, the gyroscope sensor returns back an X (Pitch) , Y (Roll), and Z (Yaw) value. Each of these values represent an axis of rotation. They are also known as Aircraft principal axes. To determine the accuracy of the sensor, we mapped the values to a 3D simulation of an airplane. The rotation of the gyroscopic sensor was replicated onto the 3D model.

Overall, we can draw conclusions about the cause of the margin of error with these sensors thanks to concepts of physics. When calculating measurements of projectiles in physics problems, physicists usually assume no air drag. When we try to calculate measurements with the sensors, we can't take out these influencing factors which leads to some variation in the data we collect.

VIII. Summary and Conclusion

After each stage of testing of both hardware and software, our results overall showed accurate and consistent readings from all of the sensors that were implemented: the ultrasonic, the TCS34725 color sensor, and the gyroscope and accelerometer sensors. Only after having solid results for each stage did we move onto the next stage. Through thorough testing during each phase of the project, we ensured that the ultrasonic distance sensor was very accurate, even after taking into account the possible margin of error. Not only that, but the color sensors were also accurate and we managed to display those through the LCD as well. The gyroscopic sensor was probably the hardest sensor to debug and test due to needing extremely accurate results, but after some time the accuracy was fixed and the results were displayed properly on the LCD.

Looking back on the project, a lot of progress has been made from the project's first conception. Reflecting on the success of the finished product, it is a very good prototype for what it can accomplish and be applied to given that this was a two quarter project. However, this project is not without its shortcomings. All of the sensors have potential faults as some have varying margins of error or require specific conditions to fully be accurate, which is a small limitation of the current prototype of the Metrigloves. While testing each product, we made the decision to ignore certain negligible factors depending on the sensor. As aforementioned, the color sensor has a fault of being slightly off in perception compared to the human eye, something that could be fixed with a hardware upgrade or better lighting. The same could be said for the distance sensor as the accuracy of the product could fade over time or there could be faulty wiring that might cause the sensor to fluctuate or skyrocket in values. Though minor flaws, these are problems with the current version of the glove and should be looked into if this project were to ever be expanded or developed further, placing an emphasis on the accuracy of the measurements. Although mainly ignored in the original design and testing phases, these problems could still pose a problem in the future if those scenarios do end up happening.

Acknowledgments

This project would not have been possible without the funding for materials from the Henry Samueli School of Engineering, along with guides from howtomechatronics.com for both how to correctly assemble the hardware along with how to program the hardware to correctly read inputs from the respective sensors. It was through the funding of the engineering school that some of the hardware for this project was gifted, thus allowing the software being worked on to be tested and integrated with the project's design.

This project would also like to give its thanks to Professor Stuart Kleinfelder for not only his generosity in organizing the class and willingness to order parts for the students, but also for being a constant reassuring and helpful resource for each step of the project. If it were not for the additional funding or guidance for the necessary steps in a successful project, there would have been a much more difficult time in creating a successful project. This project would also like to give its thanks to each and every TA in the course. Without their weekly guidance or insight on the project's development, the project's flow and pace of work would not have been as smooth. Their constructive criticism and input helped us realize design choices we shouldn't or should make for the best possible project.

In terms of the group members, credit is given to Jasper Lam's work for assembling the hardware for the ultrasonic and color sensors for the glove and also for coding it, while also obtaining the extra parts not funded through the School of Engineering. Raymond Yu's efforts are acknowledged through assisting in some of the general project management tasks of revision, proofreading, and parts researching, along with overseeing some of the hardware configurations and soldering for things like the gyroscope sensor and helping transport the hardware onto the glove. Gabriel Quach's contributions of hardware assembling and coding other aspects of the glove's components such as the LCD screen and real time clock, while also serving as the team's general project manager for deadlines and goals to meet are worth mentioning here as well.

References

[1] "UNITED STATES DEPARTMENT OF LABOR," "Commonly Used Statistics | Occupational Safety And Health Administration". *Osha.Gov*, 2017, https://www.osha.gov/oshstats/commonstats.html.

[2] *Back Injuries In Construction And Other Industries*. 2019, https://www.cpwr.com/sites/default/files/publications/CB%20page%2048.pdf.

[3] "Object Detection | Tensorflow Lite". *Tensorflow*, 2019, https://www.tensorflow.org/lite/models/object_detection/overview.

[4] Dejan, "Tutorials Archives," *HowToMechatronics*. [Online]. Available: https://howtomechatronics.com/category/tutorials/.