EE/CprE/SE 491 - sdmay19-31:
Multi-Purpose Automated Robotic Mixer (mpARM)

Week 14 Report
April 6– April 24
Client: Alexander Stoytchev/Brett Altena
Faculty Advisor: Alexander Stoytchev

Team Members

Drew Caneff — *3D Printer Specialist/CAD Designer/Accountant*
Amos Hunter — *Electromechanical Specialist/Meeting Scribe*
Brett Altena — *Meeting Facilitator/ Computer Vision Developer*
Kristian Wadolowski — *Report manager/Front-end programmer/Computer vision Developer*
Jase Grant — *Embedded Systems/ Assignment Manager*

Summary of Progress this Report

- **Frame-** There was a piece of aluminum that needed to be cut to the right size and then mounted to the frame for the camera. The camera was mounted with the plastic hardware we bought. And attached to the frame with an adhesive pad that came with the camera accessories
  In the process of getting the arm to move and testing it, we discovered that we needed to remove the frame to safely troubleshoot the arm's movements, so I unscrewed the frame from the base for now.
- **Arm-** The team took apart the Thor Arm to see what each articulation is doing when the Asgard program sends a signal to the motors. The stepper motors move for .5 seconds when we send a signal, stop moving, and then move for .5 seconds when the motion is about to end. This occurred on articulations 1, 2, and 3. There are two potential issues that could be the cause of this issue. First, since our version of the Thor Arm is not using the bump sensors or optical sensors due to issues in mounting them, the code may not be performing as expected. The code may be expecting a 0 or a 1 based on whether the sensor is being triggered at the moment or not. The next issue is that the stepper motors are mechanically not working as expected. These scenarios are going to be tested by purchasing two new stepper motors from Amazon and connecting them to our current PCB. If the motors act in the same way, our team is planning on developing an Arduino code that simply moves each articulation's stepper motor by themselves. This may be used to develop a final version of the code where we send our own signals to the robotic arm.
- **Arm UPDATE-** After rigorous testing the issue with the arm was isolated. The acceleration setting in the code was set to high and causing the motors to fail. After adjusting the acceleration setting the motors were able to move. Though additional work is needed to make the arm move in any useful way.
- **Computer vision-** Testing the computer vision code proved that 4 out of the 7 videos worked as expected. The expectation was that the code would output a signal when the pancake was ready to flip, and this corresponded with when the pancake was flipped in the video. During the other videos, the camera was not able to detect all the bubbles accurately and did not output a signal, it counted too many bubbles too soon, or it did not detect the pancake because it was not circular enough. Troubleshooting is needed in order to achieve the 80% success rate desired. Specific changes to the code will be the following: changes to how the code detects the pancake as a whole (making sure that the circle it detects is large enough to be a pancake), changing how many bubbles are required for each size of pancake, and modifying how big a bubble should be before the program counts it.
- **Camera-** The GoPro stopped working two weeks ago by becoming unresponsive to attempts of charging or utilizing it. After a phone call to GoPro, a replacement charging cable was sent to the team leader, and

the GoPro remained unresponsive. The GoPro is now going through the process of being sent back in order to have a replacement camera sent back. Currently, the team is planning on using the Team Leader's personal GoPro Hero 5 Black in case the replacement is not sent quick enough. However, the firmware is not easily supported on the FPGA, so that change in plans will need to be accounted for. Additionally, while attempting to use the camera while running the computer vision program, it defaulted to using the integrated camera on the laptop. Research will need to be done on how to set the computer vision program to use external cameras.

- FPGA- Bought a logitech c270 camera so that it can be integrated into the FPGA.
  Bitstream is a hardware pipeline to bypass some of the code so that it runs faster.
  PYNQ framework is all in python and i know some python, still learning more python.
  The way python is ran on PYNQ is different so i have to change some of the code to suit PYNQ

Pending Issues

- Complete Computer vision code
- Fix the arm
- Integrate all components

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Drew Caneff	➢ Completed poster ➢ Troubleshooted the arm ➢ Designed a user interface ➢ Performed maintenance on the arm	27	293
Amos Hunter	➢ Started slideshow ➢ Started final report ➢ Worked on frame ➢ Mounted camera	15	224
Brett Altena	➢ Performed troubleshooting on arm ➢ Reached out for outside help with troubleshooting ➢ Tested computer vision program on new footage ➢ Sent GoPro in for replacement	23	236
Kristian Wadolowski	➢ Created software solution to arm issue ➢ Compiled weekly reports ➢ Began work on final report ➢ Troubleshooted the arm	21	173
Jase Grant	➢ Successfully ran computer vision code on FPGA ➢ Acquired new camera ➢ Debugged Computer vision code ➢ Working to accelerate bitstream in FPGA	16	168

Plans for Upcoming Reporting Period

No further report periods

Gitlab Activity Summary

```
-----------------------------------------------------------
Action: joined, Tue Sep 04 2018
Author: dvcaneff
-----------------------------------------------------------
```