University of Jordan

Embedded Systems Lab

Spring 2019/2020

Project: A Simple 3DOF Robotic User-Controlled Arm

Prepared by Dr. Ashraf Suyyagh

Preferred Group Size	Grading	Project Due Date
3 Three is the **maximum and preferred** group size.	**20** Twenty marks is the project weight.	**TBD** given the recent curfew and social distancing orders **But Please start ASAP, be READY!**

Project Details and Description

In this project, you will be using a PIC16F877A to build, implement, and control a very simple 3-DOF robotic arm similar to the one show in Figure 1.

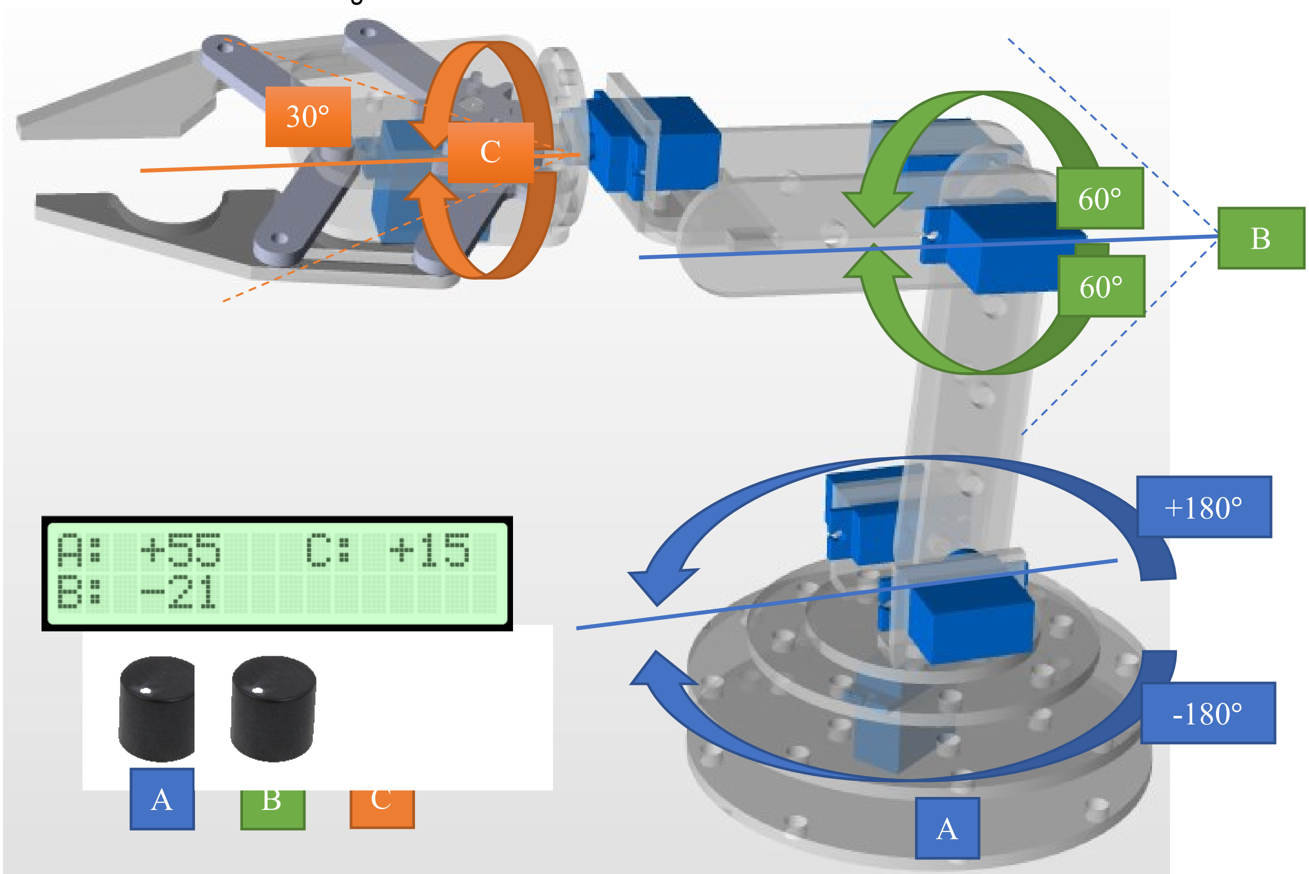

Figure 1- A sample design of the robotic arm

The robotic arm base (A) must be able to move in 180 degrees either way. That is; cover the whole circular base. The robotic arm handle (B) must be able to cover a range of -60° to 60° from the horizontal as illustrated in the figure. The clamp motor (C) must only move by a maximum of 30° along its axis. When the clamp motor angle is 0°, the clamp is closed, when it is 30°, the clamp is open to its maximum physical range.

At start-up, motors (A) and (B), must be repositioned at angle 0; and motor (C) must be in a position such that the clamp is closed. Further, an LCD shows the exact angle at which each motor is positioning the robotic arm within its respective expected range. A plus or minus sign precedes the angle for further clarity. The LCD output must match the format shown in Figure 1. You can also add the degree symbol (°) if you wish next to the angle. It is readily available in the HD44780 table. **Only show the degrees as integers for simplicity.**

The robotic arm is handled in a free-control mode by three potentiometers (control knobs). Each knob corresponds to a single motor. Each potentiometer will have a voltage range from 0 to +5 volts. Now, it is your task to convert this voltage range to the maximum possible angle of the controlled motor. Consult Table 1 below for some possible examples. Notice that at the extremes of 0 and +5 volts, each motor has a different minimum and maximum angle. Also notice that motor C only moves one-way, thus covering a positive angles range.

Similar to the procedure in the ADC experiment labsheet, and the ADC experiment video, you can draw a linear relationship between the voltage and desired angle output. You can then derive the equation which relates the ADC channel input to the desired output.

Table 1 – Voltage to Angle Degrees Transfer Points

Channel Voltage	*Motor A*	*Motor B*	*Motor C*
+5 Volts	+180°	+60°	30°
+3.75 Volts	+90°	+30°	22.5°
+2.5 Volts	0°	0°	15°
+1.25 Volts	-90°	-30°	7.5°
0 Volts	-180°	-60°	0°

Now, in this experiment you can use servo motors which you can easily find and use in Proteus simulator. **Use the hobbyist version which has three input lines**. The servo motor three lines correspond to power, ground, and the servo control signal. ***By right-clicking on the servomotor in Proteus, you can set the maximum and minimum angle ranges.*** This is important given that each of our motors has a different physical range for movement.

Now, the position of the servo motor is directly proportional to the (Pulse Width Modulated) PWM control signal. In PWM signals, the larger the pulse width, the larger the voltage/current delivered to the motor, and therefore the adjusted angle. See Figure 2 which illustrates the general concept for PWM signals and servomotor control. Usually a 50Hz (20 ms period) signal is used to control a hobbyist servomotor. We will explain how to generate PWM signals in an upcoming video. So please refer to the video for details.

In this experiment, we advise you to use hardware timers and GPIOs to generate these signals to control the three pins. Using hardware timers will ensure a smoother operation and response. You can add an oscilloscope In Proteus to test the PWM signals generation and see how they look like going from the pin towards the servomotor. This might prove useful in testing and debugging the project.

So basically, once you derive a relationship between the input voltage to the A/D and the angle, you must find a relationship between the target angle and the pulse width in the PWM signal. Alternatively, you can do this in one step; find the relationship between the A/D input, and the PWM pulse width which gives you the correct angle. We leave this part to you to figure out on your own.

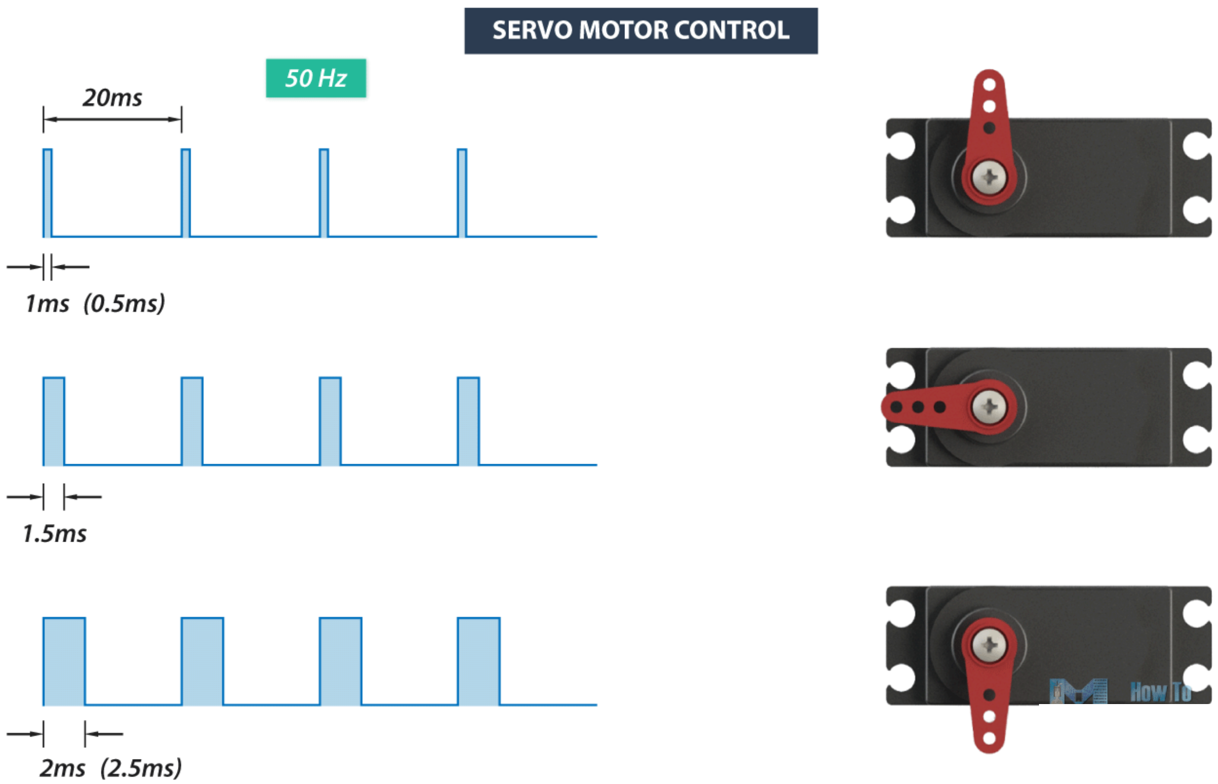

Figure 2 - PWM control signal and servo motor control (the times shown are arbitrary examples)

Project Programming Language and Simulation

Due to the current events the world is going through, the imposed curfew, and social-distancing orders, we have cancelled the hardware part. Instead, you will rely on Proteus simulation only. If you don't have Proteus software, please contact us immediately. In Proteus, you can find all parts that you need, including the servomotors, as well oscilloscopes. Please review the PDF file we posted earlier on how to use Proteus.

You are required to submit an **ASSEMBLY** code for this project. Any C submissions will be given a straight zero right away even if it is fully working. Any assembly code derived form a C compilation will be easily detected and given a grade of zero.

Project Groups

You must form groups of three to work on this project. As you can see, the project can be divided into at least three major parts, each group member can take on one part

1. The A/D programming and A/D to angle equations derivation and coding.
2. The hardware timers' part and how to generate PWM signals that correspond to the target angle.
3. LCD, system integration (putting everything together), and testing in Proteus

We prefer that you form groups from within your own lab session, but if you can not find a partner, then you can form a group with colleagues from the other sessions. Please use this Google sheet Embedded Lab Groups to register your group before **April 5th , 2020. Use the lab Facebook group to reach out to your colleagues.**

WE ARE NOT RESPONSIBLE IF YOU CANNOT FIND PARTNERS, OR IF YOUR PARTNER IS NOT GIVING ENOUGH EFFORT, OR DROPS THE COURSE!

IMPORTANT
One of ABET's assessment criteria demands that team projects have a team leader. Thus, each group must elect a group leader. The group leader is responsible for the project management, supervising the other partners, and that all team members are equally involved. The team leader is responsible for conducting meetings and ensuring professional communication between the team members, as well as the timely submission of the project when it is due (ensuring to book a timeslot for the group for discussion). **If the team leader fails to book a timeslot for discussion for his group. The group will be penalized 15% each passing day after the deadline.**

GRADING FOR CODE EFFICIENCY AND GENERALITY

It is of utmost importance to write codes which are minimum in size and execute quickly. We ask you to use subroutines, modular design and functional reuse whenever possible. In some cases, the use of indirect addressing (FSR and INDF) can be helpful in reducing code size and increasing speed**. It is important to not forget to use functional comments.**

During the discussion, we will ask you to show us on MPLAB or Proteus how your code works under different conditions so be prepared for the generic case.

The instructors and engineers are in no way responsible for helping you form groups or solve issues if your partner(s) drop(s) out from the course or is not working at all on the project. Choose your partner wisely. You might end up doing the project by yourself. **We will accept no excuses.**

Important Notes

- Start as early as possible on your project, though the project description sounds simple, there is inherent complexity in both hardware and software aspects, so do not underestimate the time it needs, you will have many problems along the way which you will have to resolve!
- Never think of buying a model or commissioning someone to do it for you, not only will you get a zero in the project, but also your act will be considered as a direct violation to JU laws and your actions shall be reported as cheating in the final exam!

- **Code sharing between groups is NOT allowed and leads to 0 points.**
- If you acquire a ***part*** of your software from a book, website, etc, then kindly reference it properly, else it will be considered as plagiarism.
- You are only allowed to base your project on PIC16877A.
- Your submitted work must be professional:
  - Software: your work should be fully documented, all inputs/outputs should be listed, and each subroutine/macro should be fully documented! Use functional comments! Refer to the last section in Experiment 2 regarding documentation.
- Divide the work such that each student is responsible for a specific task, **YET EVERY** student is required to answer for **ANY QUESTIONS** in relation to any submitted work of the project.

Report Guidelines

You should submit a hard copy of your report and it should contain the following parts:

- **Introduction**
  In this section, you give a brief description of the overall project in your own technical language.
- **System Description**

  Here you give a detailed description of the system design and how it was decomposed into subsystems.
- **Hardware System**

  Here you must explain and elaborate on your system hardware design, its inputs and outputs. Present a clear and professional circuit design schematic. **Justify why you configured and initialized your system and modules in this way.**
- **System Testing and Results**

  Present the complete methodology which you have undertaken to perform unit testing, system integration, and the final overall test. You must present the test cases which you have used to test your system for correct functionality. You must justify the choices for your test cases and whether or not they cover all possible regions of operation of your system. Provide technical discussion of the results or any abnormal operation you have witnessed during testing. **See the grading sheet for more details**
- **Conclusion**
  Give a short summary of the project, your work, and the steps you have undertaken during the design process. Furthermore, you must clearly state the contribution of each student in the project. Discuss the major obstacles that you faced during the design process.

Good Luck and Have Fun Doing the Project