Assignment: Building a Robot Arm

December 14, 2020

1 Introduction

In this coursework, you will build a simple robot arm and control interface capable of basic pick and place tasks. The arm should use three servomotors for actuation, two for moving the arm itself, and one for moving its gripper. It is also designed to be controllable through a simple control interface, consisting of two knobs for adjusting the arm postion, and a button to open and close the gripper.

2 Background

Materials

Make sure you have everything you need to complete this assignment.

The hardware items should be included in your lab kit.

The following software should be installed on your computer: (i) a terminal program (*e.g.*, `xterm` or similar on Linux/Mac OS, `cygwin` on Windows), (ii) a text editor (*e.g.*, `gedit` or similar on Linux/Mac OS, `notepad++` on Windows), (iii) `avr-libc` (including `avr-gcc`, `avr-objcopy` and `avrdude`) and (iv) `scopy`.

Item	Qty
Orangepip Kona328 Development Board	1
USB Type A - B cable	1
Breadboard	1
TowerPro SG90 Micro Servo	3
Rotary Potentiometers	2
Push Button Switch	1
Lollipop Sticks	10
Double sided tape	1
Cotton Reels	3
Jumper wires	20

Preparation

Use the items in your lab kit to build the robot arm and its control panel. An instructional video on how to build the arm is provided on KEATS. You are permitted to make minor alterations to the mechanical design provided your robot follows the same basic layout of two rotary joints and one gripper. Use the diagram in Fig. 1 to build the circuit connecting the arm to the development board. To aid debugging, you may choose to do this in stages as you work through the tasks in §3 instead of completing the entire build in one sitting.

To complete the assignment, you will need the following files:

- `robot.c`
- `librobot.h`
- `librobot.c`
- `k0000000.c`

Download these from KEATS and save them to your computer.

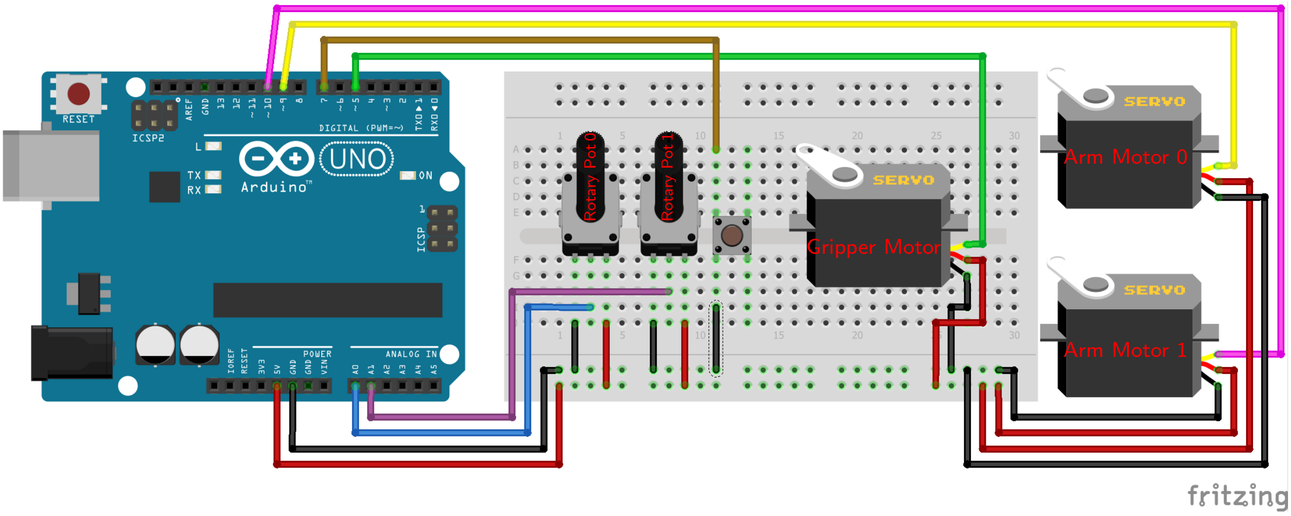

Figure 1: Circuit.

Use the following commands to compile the program and download it to your development board:

```
$ avr-gcc -Wall -g -Os -mmcu=atmega328p librobot.c k0000000.c robot.c -o robot.elf
$ avr-objcopy -O ihex robot.elf robot.hex
$ avrdude -p atmega328p -c arduino -P COM8 -b 115200 -D -U flash:w:robot.hex:i
```

You should find that it compiles without errors. Adjust the port settings in the avrdude command if necessary to enable the program to be downloaded to the development board [1]

3 Tasks

Complete the following tasks.

1. The first task is to set up the push button to control the action of the gripper motor.

   To complete this part of the assignment, find and modify the following lines in the template file:

   ```
     /* --------------- Begin Answer to Task 1a Here ------------- */

     /* --------------- End Answer to Task 1a Here --------------- */
   ```

   and

   ```
     /* --------------- Begin Answer to Task 1b Here ------------- */

     /* --------------- End Answer to Task 1b Here --------------- */
   ```

   Implement code in the function init_button_sensor() to configure the appropriate digital pin of the development board as an input, so that it can detect when the push button is pressed. The function should make appropriate use the Atmega328P's internal pull-up resistors to ensure that the state of the pin is well-determined at all times.

   Implement code in the function read_button_sensor() to read the state of the appropriate digital pin of the development board. The function should return 0 if the button is pressed, and 1 otherwise.
   [3 marks]

[1] Your compiler may issue warnings about unused variables. These can be safely ignored.

2. The second task is to set up the rotary potentiometers as control knobs to set the desired position of the arm's main motors.

   To complete this part of the assignment, find and modify the following lines in the template file:

   ```
   /* ---------------- Begin Answer to Task 2 Here ------------- */

   /* ---------------- End Answer to Task 2 Here --------------- */
   ```

   Implement the function `read_knob_sensor()` to read the position of the rotary potentiometer using the appropriate ADCs of the development board. The function should take an integer value `knob` as an input, where `knob` can take the value `0` if a reading from `knob0` is required, and `1` if a reading from `knob1` is required. It should return a floating point value corresponding to the position of the knob between zero (*i.e.*, knob at minimum position) and one (*i.e.*, knob at maximum position).

   [4 marks]

3. The final task is to write code to configure the control of the main arm motors through pulse-width modulation.
   To complete this part of the assignment, find and modify the following lines in the template file:

   ```
   /* ---------------- Begin Answer to Task 3 Here ------------- */

   /* ---------------- End Answer to Task 3 Here --------------- */
   ```

   Implement the function `init_arm_motors()` to configure Timer1 of the Atmega328P to enable pulse-width modulation with the following characteristics: (i) it should enable the generation two PWM waves from the same timer in *Fast PWM* mode, (ii) the PWM waves should have a period of $20\,ms$, (iii) it should be possible to adjust the duty cycle to implement pulses between $1\,ms$ and $2\,ms$ at a resolution of $0.5\,\mu s$.

   [8 marks]

4. *Optional task*. Produce a video demonstrating you controlling your robot arm to stack the cotton reels supplied in your lab kit. The task should start with the cotton reels placed side-by-side on a flat surface, and end with the reels placed one on top of another, with green at the bottom, yellow in the middle and red at the top.

   Two *bonus marks* are available for this task at the discretion of the marking team. Marks will only be awarded for videos that clearly show that (i) it is you operating the robot, (ii) the hardware has been built by you, (iii) the software running on the robot is the same as that submitted for Tasks 1-3 above. Videos submitted for this task must be in MP4 format (see submission instructions below).

   Note that, total marks for the assignment are capped at a maximum of 15 marks, so bonus marks will not be awarded for this task to those who score full marks on all other tasks in the assignment.

4 Submission

To submit your assignment, you need to upload your code to KEATS. To do this, follow the below instructions.

First, change the file name of `k0000000.c` to your k-number. For example, if your k-number is k1234567, the file name of your assignment should be `k1234567.c`. Next, submit this file to KEATS by following the *Assignment 2* link.

Important The code you submit will be assessed based on its compatibility with the specifications, including whether it runs with the template code provided. *If it does not run, it may be awarded zero marks.*

If you upload a video submission for the *optional task* (Task 4), your video must be in MP4 format, with file name corresponding to your k-number. For example, if your k-number is k1234567, the file name of your video should be `k1234567.mp4`. This should be submitted alongside your main submission on KEATS by following the *Assignment 2* link.

Completed assignments should be submitted to KEATS by 5pm (UK time) on January 11, 2021.

This assignment is worth 15% of the module mark.