Assignment 5

Roberto Solis, Carson Wagner

April 5, 2024

1 Introduction

The objective of this lab assignment is to use to LPC1769 Microcontroller alongside with the analog joystick to create a controller for a hobby servo motor. We will do this by interfacing the joystick's x-axis potentiometer with the LPC1769's analog-to-digital subsystem. We also need to generate a PWM signal with a variable pulse width that will be between 1ms and 2ms which will correspond to the position being 0 degrees to 180 degrees of the joystick.

2 Software Code

```
/*
 * Copyright 2022 NXP
 * NXP confidential.
 * This software is owned or controlled by NXP and may only be used strictly
 * in accordance with the applicable license terms.  By expressly accepting
 * such terms or by downloading, installing, activating and/or otherwise using
 * the software, you are agreeing that you have read, and that you agree to
 * comply with and are bound by, such license terms.  If you do not agree to
 * be bound by the applicable license terms, then you may not retain, install,
 * activate or otherwise use the software.
 */

#ifdef __USE_CMSIS
#include "LPC17xx.h"
#endif

#include <cr_section_macros.h>


#define PCONP (*(volatile unsigned int *) 0x400FC0C4)
#define PCLKSEL0 (*(volatile unsigned int *)0x400FC1A8)

#define AD0CR (*(volatile unsigned int *)0x40034000)
#define AD0GDR (*(volatile unsigned int *) 0x40034004)

#define PINSEL1 (*(volatile unsigned int *)  0x4002C004)
#define PINSEL4 (*(volatile unsigned int *)0x4002C010)

#define PWM1TCR (*(volatile unsigned int *) 0x40018004)
#define PWM1MR0 (*(volatile unsigned int *) 0x40018018)
#define PWM1MR1 (*(volatile unsigned int *) 0x4001801C)
#define PWM1MCR (*(volatile unsigned int *) 0x40018014)
#define PWM1PCR (*(volatile unsigned int *) 0x4001804C)
#define PWM1LER (*(volatile unsigned int *) 0x40018050)


float dataADC;


void initializeConvert() {

  // Setting the A/D Converter power/clock control bit (Turning A-to-D on)
  PCONP |= (1 << 12); // Setting the A/D Converter power/clock control bit (Turning A-
    to-D on)
```

```
44
45    // Enable the ADC by setting PDN Bit in AD0CR Register
46    AD0CR |= (1 << 21);
47
48    // Selecting AD0.0
49    PINSEL1 |= (1 << 14);
50
51    // Set AD0.0 to Input
52    AD0CR |= (1 << 0);
53
54    // Disable Burst Bit (Set to 0)
55    AD0CR &= ~(1 << 16);
56
57
58  }
59
60
61  void initializePWM() {
62
63    // PWM Initializing as down in Lecture Slides
64    PWM1TCR = 0;
65    PCLKSEL0 |= (1<<12); // PWM PCLK = CCLK/1
66    PWM1MR0 = 80000; // 256 PCLK cycle period (8-bit equivalent)
67    PINSEL4 |= (1<<0); // Configure P2.0 as PWM1.1
68    PWM1MCR = (1<<1); // Reset counter on MR0 match
69    PWM1PCR = (1<<9); // Single edge mode and enable PWM1.1 only
70    PWM1TCR = (1<<0) | (1<<3); // Enable counter and PWM mode
71
72
73  }
74
75
76  unsigned int readConversion() {
77
78    // Start A-to-D Conversion
79    AD0CR |= (1 << 24);
80
81
82    // Wait for Done bit to go to 1
83    while(!((AD0GDR >> 31) & 1)) {
84      // Do nothing
85    }
86
87    // Clear A-to-D Start
88    AD0CR &= ~(1 << 24);
89
90    // Read Converted Result
91    int data = ((AD0GDR >> 4) & 0xFFF);
92
93    return data;
94
95  }
96
97  int ticks;
98
99  // Function to create a delay in milliseconds
100 void wait_ms(float ms) {
101
102
103   volatile float i;
104
105
106   float m = 0.002719;
107   float b = 0.1;
108
109     ms = (ms-b)/m;
110
111     for (i = 0; i < ms; i++) {
112     //do nothing
113     }
114 }
```

```
115
116
117 int main(void) {
118
119   // Initialize ADC and PWM
120     initializeConvert();
121   initializePWM();
122
123
124     while(1) {
125
126       // Conversion and Calculation for PWM
127       dataADC = readConversion();
128       PWM1MR1 = ((204.75 + 0.05 * dataADC) * 80000/4095);
129       PWM1LER = (1 << 1);
130       wait_ms(1);
131
132     }
133     return 0 ;
134 }
```

2.1 Oscilloscope Screenshots

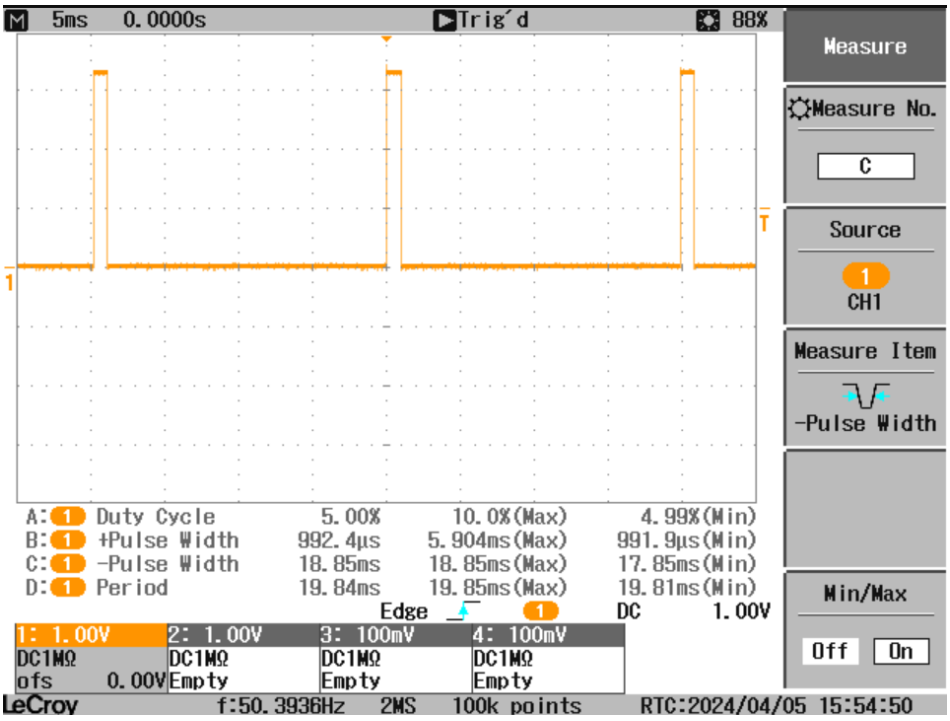


Figure 1: 1 ms at 0 degrees snapshot

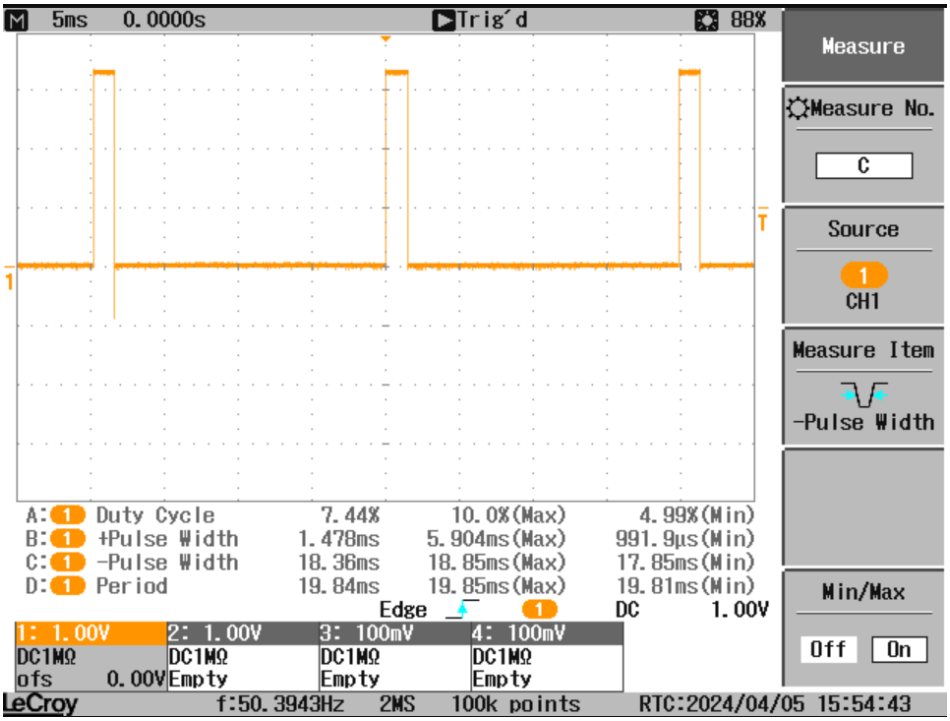


Figure 2: About 1.5ms at 90 degrees snapshot

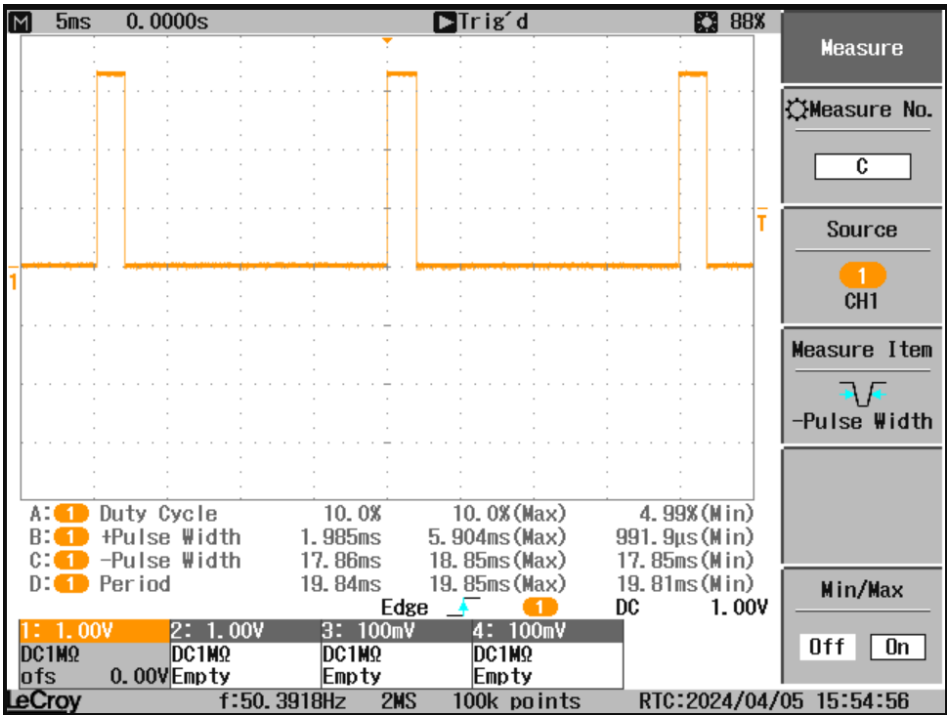


Figure 3: About 2ms at 180 degrees snapshot

3 Hardware

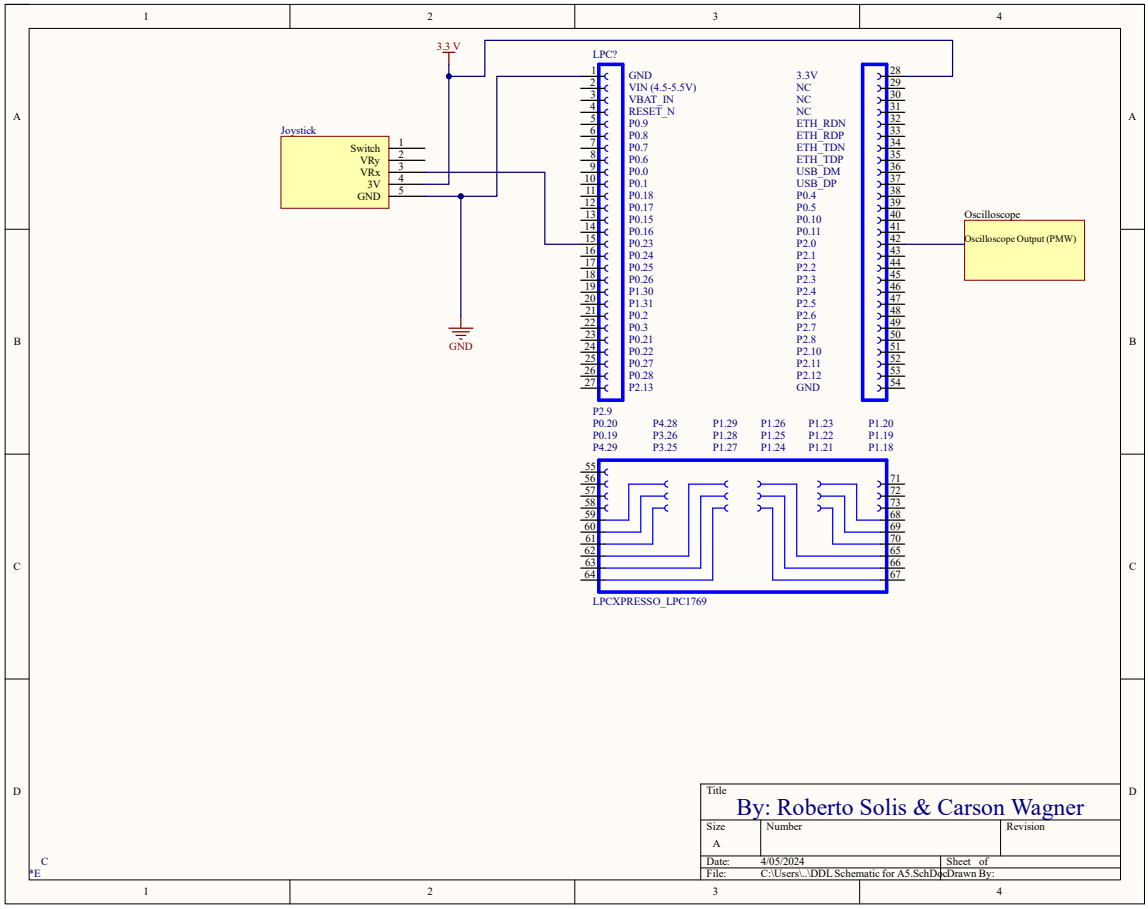
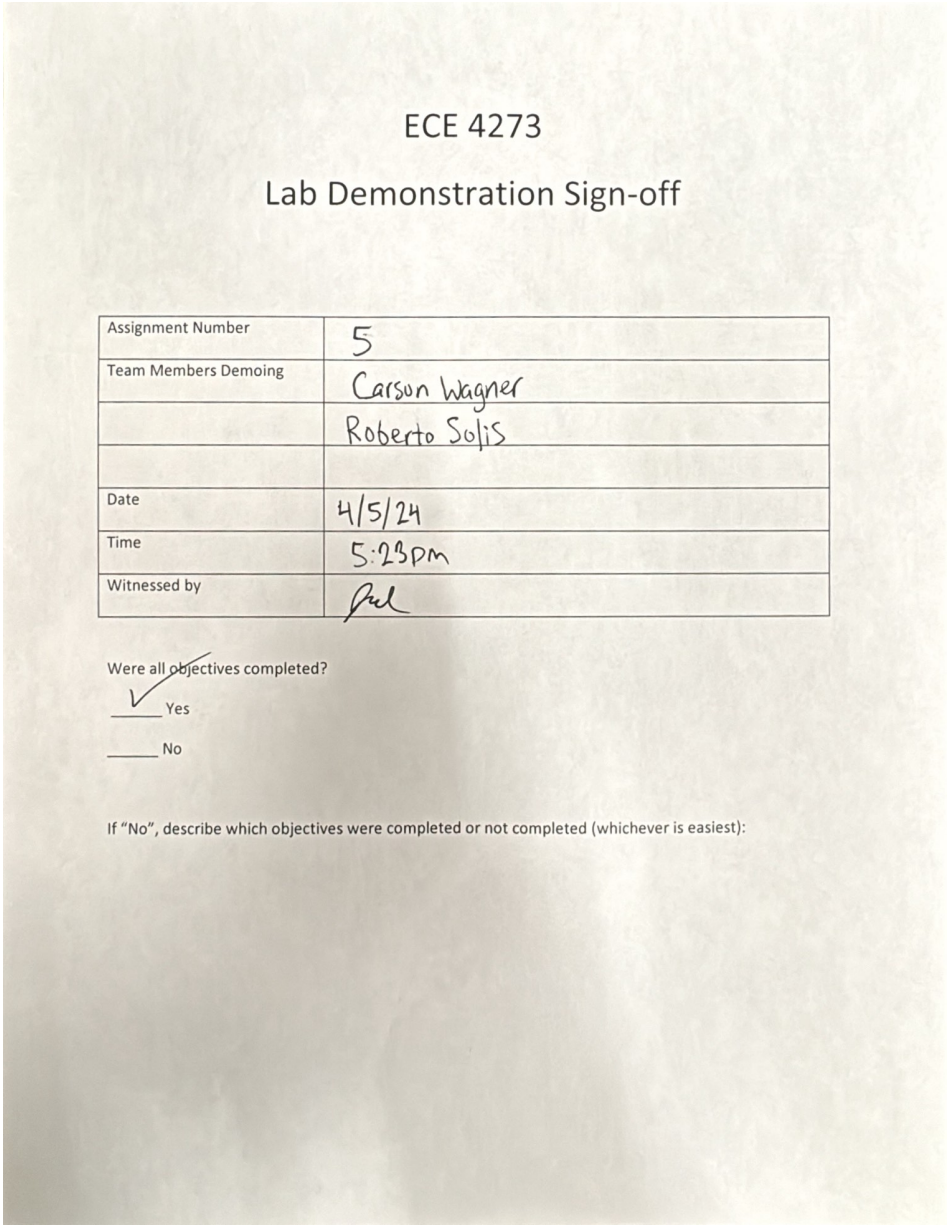


Figure 4: Hardware Schematic

4 Lab Demonstration Sheet



ECE 4273

Lab Demonstration Sign-off

Assignment Number	5
Team Members Demoing	Carson Wagner
	Roberto Solis
Date	4/5/24
Time	5:23pm
Witnessed by	

Were all objectives completed?

✓ Yes

_____ No

If "No", describe which objectives were completed or not completed (whichever is easiest):

Figure 5: About 2ms at 180 degrees snapshot

5 Contributions

Carson Wagner(Computer Engineering)

Worked on and finished the code along with helped writing the report and schematic

Roberto Solis(Electrical Engineering)

Worked on Schematic and help write report. Also helped with some code.