ECE383: Microcomputers – Lab 8
Analog-to-Digital and Digital-to-Analog Conversion

Goals: The goals of this lab are to introduce students to a PIC24-based Microstick II system capable of analog-to-digital (ADC) and digital-to-analog (DAC) conversions. The laboratory requires students to connect a SPI-based MAXIM 548A DAC to the Microstick II and program the system to act as a data acquisition and display device.

1. Introduction

This lab introduces a basic analog to digital data conversion using an ADC module on the PIC24 and digital to analog conversion using the SPI-based MAXIM 548 DAC.

The topics covered in the lab include:

- Use of timers and timer interrupts to control data acquisition.
- Analog-to-digital and digital-to-analog conversion.
- Communication with peripheral devices using SPI communication

The tasks in this lab include:

- Wiring a MAXIM 548A integrated circuit to the Microstick II on a breadboard.
- Write a C program to program the PIC24 system to act as a data acquisition and display device.

2. TASK 1: DAC wiring

Wire the MAX548A DAC into your **existing layout containing the LCD** wiring that was constructed in a previous lab as shown in Figure 1:

1. Connect all required **Vdd (3.3V)** and **Vss (GND)** pins on the MAX548A to your system power and ground rails respectively.
2. Connect the **LDAC# pin (pin 6)** to **Vdd**.
3. Connect the **CS# pin (pin 3)** on the MAX548A to **RB3 (pin 7)** on the PIC24 Processor.
4. Connect the **DIN pin (pin 4)** on the MAX548A to **SDO1/RP11 (pin 22)** on the PIC24 processor.
5. Connect the **SCLK pin (pin 5)** on the MAX548A to **SCK1/RP12 (pin 23)** on the PIC24 processor.
6. Connect the **OUTA pin (pin 2)** on the MAX548A to **RB14/AN10 (pin25)** on the PIC24 processor.

Connect the photocell and resistor (910 Ω) to **RA1 pin (AN1)** of the PIC24 processor as shown in Figure 1. A photocell is a component that changes its physical resistance based on the current levels of light. Using this component, it is possible to design systems that can measure and react to the amount of light in an environment (think a night-light that turns on automatically when it gets dark). During this lab, you will use this component to explore how it interfaces with the PIC24 to achieve these types of tasks.

Version: 3
Revision Date: 12-Nov-2020

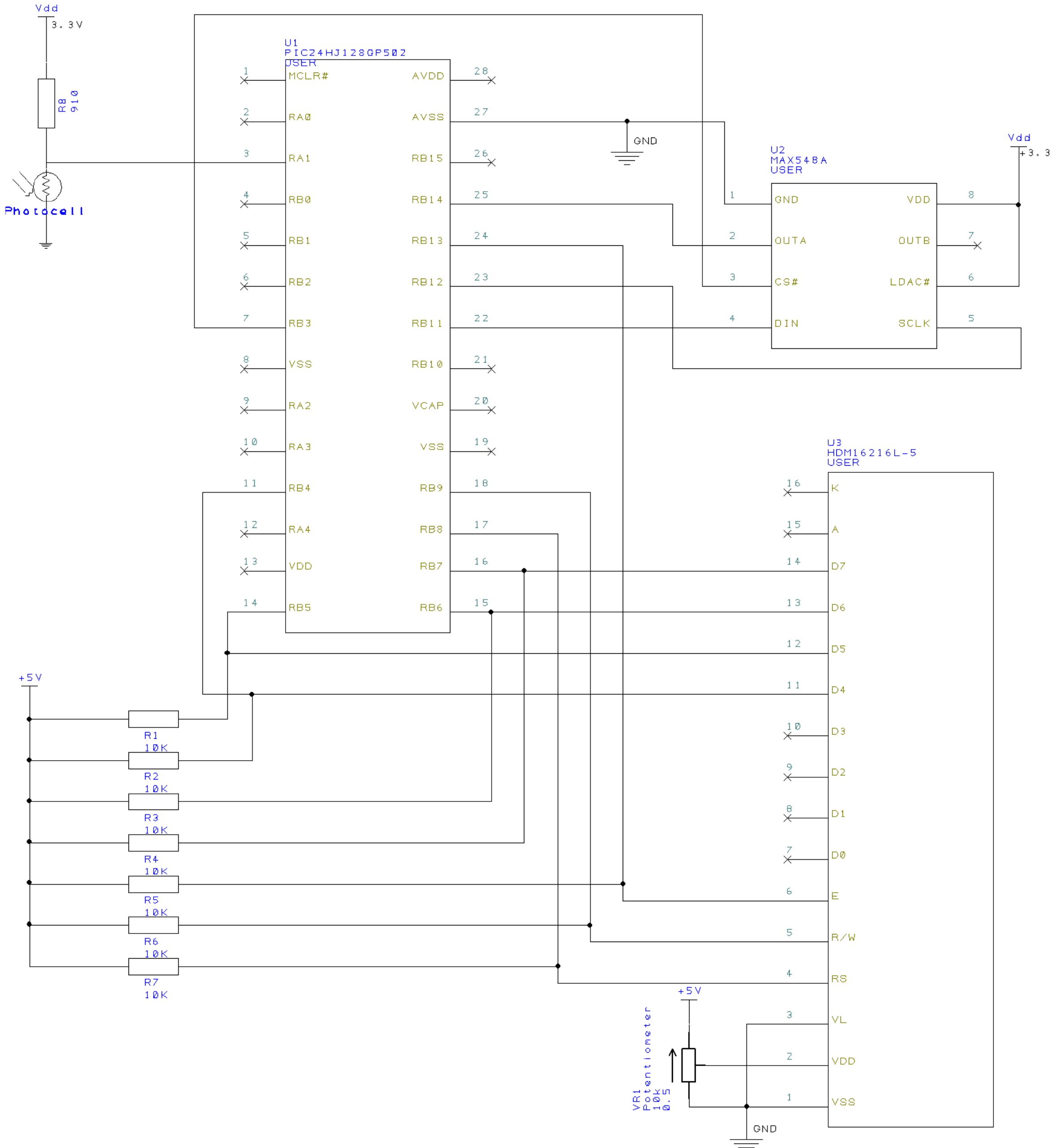

Figure 1. Task 1 schematic

Use the *chap11/adc_spidac_test.mcp* MPLAB project from the code library as the starting point for this task. Modify the pin assignment in this program to match the circuit diagram shown above (pay close attention to the connections between the PIC24 and the MAX548A). Then, compile and download it to the Microstick II for verification.

This program will use the ADC to sample a voltage, convert the digital value to an 8-bit representation, and send that 8-bit value to the MAX548A DAC using the SPI communication protocol. The MAX548A DAC will then convert that digital value to an analog voltage that will be measured by the PIC ADC. You will verify small pieces of this total process in each task building toward the complete program and functionality. For now, you are using this MPLAB project in TASK1 to confirm that the ADC and the photocell are working as expected.

If your code and system are constructed correctly, the output of the photocell / R1 voltage divider that is the input to the ADC of pin RA1 will be **3.3V** when the photocell is **fully covered** and **0V** when the photocell is **fully illuminated**. Why?!

The photocell resistance decreases when it is bright and increases when it is dark. Using this information, we can consider the expected voltages that can occur using knowledge about voltage dividers. Voltage dividers are circuits that divide a voltage (3.3V for this exercise) across multiple resistors (photocell and 910 ohms for this lab). The equation to help us understand this for the specific setup of our photocell and 910 ohm resistor is:

$$V_{RA1} = 3.3V \frac{R_{photocell}}{R_{photocell} + 910\Omega}$$

where V_{RA1} is the voltage that will be measured by the ADC on pin RA1 and $R_{photocell}$ is the resistance (in ohms) of the photocell. Using this voltage divider formula, consider what happens to the V_{RA1} if $R_{photocell}$ is very small (approaching 0 ohms), like what could be expected for a very bright situation:

$$V_{RA1} = 3.3V \frac{0\ \Omega}{0\Omega + 910\Omega} = 0V$$

Now consider what happens to V_{RA1} if $R_{photocell}$ is very large (above 100 kOhms), like what could be expected for a very dark situation:

$$V_{RA1} = 3.3V \frac{100\ k\Omega}{100k\Omega + 910\Omega} \cong 3.27V$$

Using the code that you downloaded to the Microstick II, verify this behavior by covering the photocell and then using a cell phone light to illuminate it. During each of these states, have a breakpoint set that will allow you to stop executing the code after the ADC value is captured and converted to a float value (**f_adcVal**). Next, when the photocell is partially covered, what is the reported voltage **f_adcVal?** For this partially covered value use the previously detailed equations to calculate the resistance value of the photocell ($R_{photocell}$). If completing this laboratory on campus (or if you have your own hand-held multimeter), this stage is a great moment to use a digital multimeter to measure the voltage at pin RA3 to confirm it matches what is being reported by the PIC24.

Deliverable 1: Upload your hand-calculations for the photocell resistance value that you have calculated for your partially covered case. Show all steps and values used for this calculation.

Deliverable 2: Upload a video that captures your TASK 1 program running on the PIC24 hardware (the Microstick II). This video needs to show the successful build, download, and execution of the code. During execution, this video needs to capture/report the **f_adcVal** during the covering/illumination steps to confirm all hardware/software is operating as expected. You need to provide clear audio descriptions of the testing steps as you are recording to help the instructors follow your process. At the end of your video also make sure to include a final shot that shows your ACT card or similar photo ID. Videos without this final detail will be given a grade of zero.

Deliverable 3: Upload the C-code source file (.c file) you wrote to meet the TASK 1 requirements. Remember that comments are required for all source code to detail the intent of the code. Source code without comments will be given a zero.

3. TASK 2: PIC24 Data Acquisition

While we worked to confirm that the photocell and ADC were operating as expected in TASK 1, we have not yet confirmed the DAC operation. This will require using SPI to configure the DAC as well as read the voltage output by the DAC using an ADC. In this task, you will need to write a program that alternates between which pin will be used by the ADC to record values of the photocell circuit (AN1) and the MAX548A output (AN10) and then compare these values. Specifically, your program will need to:

1. Configures both RA1/AN1 and RB14/AN10 to be analog inputs.
2. Configure AN1 to be the ADC pin to be sampled.
3. Sample the analog voltage on RA1/AN1, converts the resolution to match the resolution of the DAC (PIC24 ADC is 10 or 12 bits, DAC is 8 bits). The extra bits may be trimmed by a shift operation.
4. Sends the digital code to the DAC **OUTA pin (pin 2)** on the MAX548A.
5. Configure AN10 to be the ADC pin to be sampled.
6. Sample the analog voltage on RB14/AN10.
7. Convert the sampled analog voltage from its digital representation to a float that represents the decimal version of the measured voltage
8. Repeats steps 2-7 continuously.

The *chap11/adc_2pots1.mcp* MPLAB project from the code library presents code that configures and alternates between using two different pins for the ADC, which you may find helpful for TASK2 to learn how to configure and read two different pins. It is recommend to use your TASK1 project/code as a starting point, and import the useful code from the *chap11/adc_2pots1.mcp* project as needed.

After you have completed your program, compile and download this program to the Microstick II for verification. Using the code that you downloaded to the Microstick II, verify that the value output by the MAX548A DAC is the expected value. That is, does the value measured on RB14 after sending the digital code to the DAC match your expectations?

Deliverable 4: Upload a video that captures your TASK 2 program running on the PIC24 hardware (the Microstick II). This video needs to show the successful build, download, and execution of the code. During execution, you need to show that the measured DAC value on pin RB14 meets the expectations based on the digital value sent to the MAX548A. Use appropriate break-points in your code and the watch window to report the variable values to validate your program operation. You need to provide clear audio descriptions of the testing steps as you are recording to help the instructors follow your process. At the end of your video also make sure to include a final shot that shows your ACT card or similar photo ID. Videos without this final detail will be given a grade of zero.

<u>Deliverable 5:</u> Upload the C-code source file (.c file) you wrote to meet the TASK 2 requirements. Remember that comments are required for all source code to detail the intent of the code. Source code without comments will be given a zero.

4. TASK 3: PIC24 Data Acquisition and LCD Control

Next, to further develop mastery of using the ADC, DAC, and manipulating digital values you need to write a program that:

1. Samples the analog value Vin on the AN1 input every 0.01 seconds. <u>You must use a timer-driven interrupt service routine to control the input sampling</u>.

2. Convert the resolution of the sample to match the resolution of the DAC (PIC24 ADC is 10 or 12 bits, DAC is 8 bits). The extra bits may be trimmed by a shift operation.

3. Send the digital code to the DAC, use **OUTB pin** on the MAX548A (this is different than **OUTA** that was used in the previous tasks). Refer Page 9 of the MAX548 datasheet to see how to do this. The datasheet is available on the class website. This step is to get further practice in reviewing the datasheet details and how to control the MAX548A over SPI.

4. Convert the binary value acquired by the PIC24 ADC to a brightness in the range 0-255 and output this value to the LCD. When the photocell is fully covered, the LCD screen should display 0. When the photocell is fully illuminated, the LCD screen should display 255. Use this equation to scale the input.

   Vscaled = 255 * (Vin-Vinmin)/(Vinmax-Vinmin)

5. When the ADC voltage Vin is **below 1.7V**, **turn on** an LED (with a 910 ohm series resistor) that is connected to the RA0 pin on the PIC24 (you will need to add this hardware, it was not originally included for task 1). When Vin is **above 1.7V**, **toggle** the LED at a rate of 10 times per second.

Compile and run the MPLAB project to verify operation of your system.

<u>Deliverable 6:</u> Upload a video that captures your TASK 3 program running on the PIC24 hardware (the Microstick II). This video needs to show the successful build, download, and execution of the code. During execution, you need to clearly show that the LCD is operating as expected (value of 0 when photocell covered, value of 255 when illuminated, and what the value is based on the ambient light in the environment) and that the LED is operating as expected (LED turned on for voltages < 1.7V and toggling at 10 times per second for voltages > 1.7V). You need to provide clear audio descriptions of the testing steps as you are recording to help the instructors follow your process. At the end of your video also make sure to include a final shot that shows your ACT card or similar photo ID. Videos without this final detail will be given a grade of zero.

<u>Deliverable 7:</u> Upload the C-code source file (.c file) you wrote to meet the TASK 3 requirements. Remember that comments are required for all source code to detail the intent of the code. Source code without comments will be given a zero.