# Lab 1 Pre-lab Example

## Team Information

**Lab number:** Click here to enter text.

**Date:** Click here to enter a date.

**Team Members:** Click here to enter text.

**Team Number/Name:** Team Member Responsibilities

**Software Design:** Click here to enter text.

**Hardware Design:** Click here to enter text.

**Quality Assurance:** Click here to enter text.

**Systems Integrator:** Click here to enter text.

# Hardware

### Responsibility (2 pts)

Fill in the table below based on your responsibilities provided in the procedures and grading rubric. This will be what determines your individual grade for the lab.

|  |  |  |
| --- | --- | --- |
| Part 1 | Part 2 | Part 3 |
|  |  |  |

### Part 1 (1 pts)

Draw the schematics or create a table detailing the connections for Part 1 of Lab 1. **An example of the level of detail that is expected is given below**.



### Part 2 (1 pts)

Draw the schematics or create a table detailing the connections for Part 2 of Lab 1. You can choose to use a table or use a diagram.

### Part 3 (1 pts)

Draw the schematics or create a table detailing the connections for Part 3 of Lab 1. You can choose to use a table or use a diagram.

# Quality Assurance

### Responsibility (2 pts)

Fill in the table below based on your responsibilities provided in the procedures and grading rubric. This will be what determines your individual grade for the lab.

|  |  |  |
| --- | --- | --- |
| Part 1 | Part 2 | Part 3 |
|  |  |  |

### Part 1 (1 pts)

List the tests that you intend to do based on the Lab 1 procedures. Describe the name of the test, the tool you intend to use, and a description of the test. Do this for each part in Lab 1.

|  |  |  |
| --- | --- | --- |
| Test Name | Tool | Description |
| Continuity Test | Digital Multi-meter | Test all wire connectors, solder joints, and wire-wraps for continuity |
| Power Test | Digital Multi-meter | Test that any created circuits have power correctly flowing |
| Grounding Test | Digital Multi-meter | Test that any switches connected to ground actually ground a powered circuit |
| Component Test | Digital Multi-meter | Test that appropriate pins on the switch are connected |

You may also include any software tests that you intend to make.

|  |  |  |
| --- | --- | --- |
| Test Name | Input | Description |
| timerTick Test |  | Test that timerTick indeed ticks at the correct interval |
| displayTime Test | “10000” | Test that this function assigned the appropriate register to “10:00:00.” |
| Register Test |  | Test that the register configurations for the timer work. |

### Part 2 (1 pts)

### Part 3 (1 pts)

# Software

### Responsibility (2 pts)

Fill in the table below based on your responsibilities provided in the procedures and grading rubric. This will be what determines your individual grade for the lab.

|  |  |  |
| --- | --- | --- |
| Part 1 | Part 2 | Part 3 |
| Buttons are debounced. LEDs are clearly distinguished in the code. The RUN LED is initially on. | All functions work properly | Interrupts are used, state machine implementation, requirements are fulfilled. |

### Part 1 (1 pts)

List the relevant control registers for controlling the LEDs in Part 1 of Lab 1.

|  |  |
| --- | --- |
| Device: | Register(s): |
| Digital I/O | TRISGbits.TRISG12 = 0 (output)  TRISGbits.TRISG14 = 0 (output)  LATGbits.LATG12 and LATGbits.LATG14 to turn them on and off. |

Also describe the function of the microcontroller software as a finite-state machine.

The software FSM will have 3 states, one for Run, one for stop and one for debouncing.

Debounce

Run

**Run**: if change in button will go to Debounce

**Stop**: if change in button will go to debounce

**Debounce**: if pushed will go to other state.

If released will go back to the other state.

Stop

### Part 2 (1 pts)

List the relevant control registers for controlling the LCD in Part 2 of Lab 1.

|  |  |
| --- | --- |
| Device: | Register(s): |
| Timer | TMR2 |

### Part 3 (1 pts)

Also describe the function of the microcontroller software as a finite-state machine in Part 3 of Lab 1.

**Run**: if change in button will go to Debounce

Else will count

**Stop**: if change in button will go to debounce

Else will retain value

**Debounce**: if pushed will go to other state.

If released will go back to the other state.

Stop

Debounce

Run