# Network-Enabled Sump Pump with Email Alert

Seth Dynes ECET 491

Course Professor

Paul I. Lin, Professor of ECET 491 Senior Design II


1

## Table of Contents



2

## Overview

- This system will allow for semi remote monitoring of the sump pit in basements that are prone to flooding.
- Pump failure, heavy rain, a plugged outlet, a power outage or just a rain in the middle of the night all are reasons a basement can flood.
- This system will have audio, visual, and electronic alarms.

3

## Purpose

- My system will monitor how much water is in the pits and alert the home owners if they rise above preset levels.
- This would allow the home owner to do other daily tasks around the house without have to constantly check the well pit
- When there is a lot of rain forecasted or there is a risk of popup thunder storms during the night the system will bring home owners peace of mind.

4

# Deliverables

- Prototype – The device will meet the scope definition goals.
- Presentation –Present a summary of project activity, and demonstrate a prototype
- Report – A written report will be produced to contain timeline, project details, challenges and measurements

5

# Costs

- The total cost estimates for the project to date is $176 well bellow the initial estimates of $400.

**Material/Tool Cost**

| Item | Qty | Cost Each | Total Cost | Comments |
|---|---|---|---|---|
| ARDUINO MEGA 2560 REV3 | 1 | $ 12.38 | $ 12.38 | Arduino |
| Bluetooth / 802.15.1 Development Tools | 1 | $ 19.95 | $ 19.95 | Mouser.com |
| SENSOR DISTANCE ULTRASON PING | 1 | $ 3.99 | $ 3.99 | www.alliedelec.com |
| Circuit board | 0 | $ 15.00 | $ - | Custom made |
| Misc electronic components | 1 | $ 50.00 | $ 50.00 | Mouser.com |
| Wall Wart | 1 | $ 20.00 | $ 20.00 | |
| Enclosure and Hardware | 0 | $ 100.00 | $ - | Mouser.com |
| Automatic float and trickle charger | 1 | $ 7.99 | $ 7.99 | Harbar Freight |
| Arduino Ethernet sheild | 1 | $ 29.99 | $ 29.99 | DIY IT |
| Terminal Pin Block PA3398 | 1 | $ 3.99 | $ 3.99 | DIY IT |
| 12V 2 Channel Switch | 1 | $ 9.99 | $ 9.99 | DIY IT |
| Relay Module | 2 | $ 3.99 | $ 7.98 | DIY IT |
| Stackable Headers 6pin 5pcs | 1 | $ 2.99 | $ 2.99 | DIY IT |
| Copper PortotypeBoard pcb | 1 | $ 6.99 | $ 6.99 | DIY IT |
| Wire | 4 | $ - | $ - | Grandpa |
| Bell | 1 | $ - | $ - | Grandpa |
| 12V Battery | 1 | $ - | $ - | Grandpa |
| Terminal Strips | 2 | $ - | $ - | Grandpa |
| | | | $ - | |
| | | | $ - | |
| | | | $ - | |
| | | | $ - | |
| | | | $ - | |
| | | | $ - | |
| | | | $ - | |
| | | Material Total | $ 176.24 | |

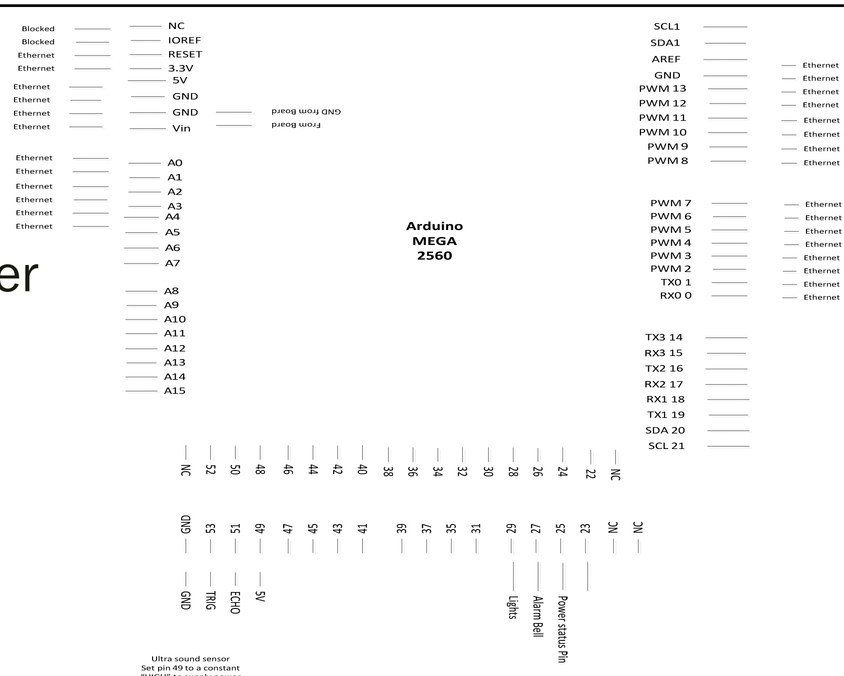
6

# Technical

- The operation of the system is very simple. A Sensor measures the water level and the program decides if you should be alerted or not. The sensor uses ultrasound to measure the water level
- Alarms come in three different types: an audio alarm, a visual alarm, and an electronic alarm. The audio alarm will be a bell. The visual alarm will be a flashing light. The electronic alarm is the text or email send to the user.
- Primary power will be from 110 AC Wall power and a secondary power from a 12V DC battery.

7

# Pin Mapping for the Arduino Microcontroller




8

# System Requirements

- Shall measure Water in the pit ---Met
- Shall have an alarm when the water hits a predetermined level- Met
- An alarm shall be audible -Met
- An alarm shall be visual -Met
- An alarm shall be sent via a network –Not Met
- Shall have a backup Battery -Met
- Shall maintain a Backup Battery -Met

9

# Arduino Mega 2560

- It also is the only Arduino Microcontroller that will accept Shields and still have open Input-Output Ports

| | |
|---|---|
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 54 (of which 15 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |
| LED_BUILTIN | 13 |
| Length | 101.52 mm |
| Width | 53.3 mm |
| Weight | 37 g |

10

# One Relay Module:

- It is designed for use with an Arduino Microcontrollers.
- The relay is optimized for low current switching even so provisions are made for the safe current labels.
- This module takes an “Active High” to the signal pin to operate the solenoid.

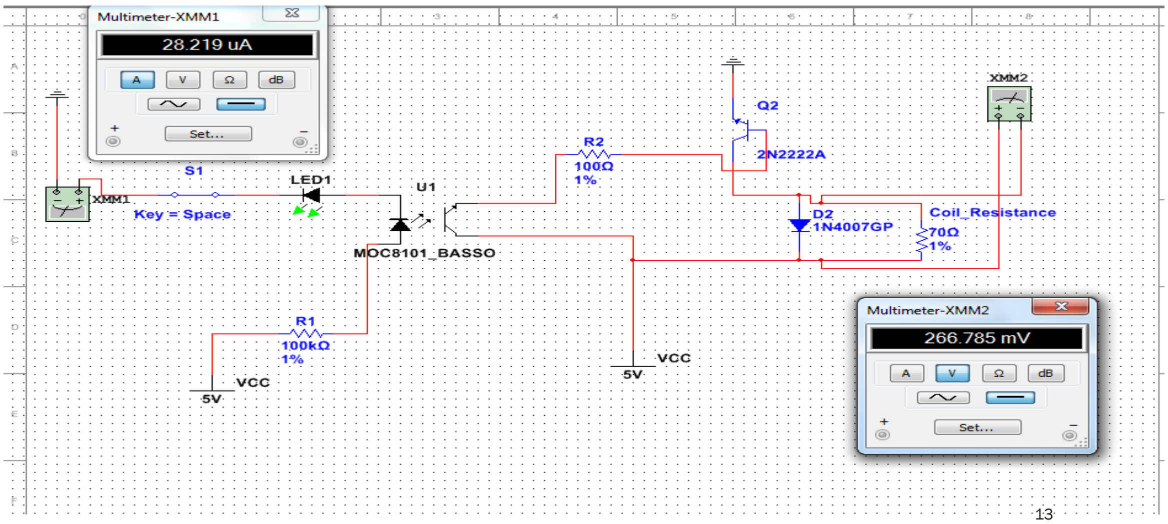
11

# Two Relay module:

- It is designed for use with an Arduino Microcontrollers.
- The relay is optimized for low current switching even so provisions are made for the safe current labels.
- This module takes an “Active Low” to the signal pins to operate the solenoid.

12

# Two Relay module:




13

# Internet Communication

- **Ethernet Shield:** The expandable board connects the microcontroller to the Internet via a Cat 5E Ethernet cable. The shield has a through hole connectors to allow another shield to be but on top of it. This means that I could stack and operate a WI-FI shield on top yet to offer a choice of Internet connection methods.
- **WI-FI Shield :** The expandable board connects the microcontroller to the Internet via a WA-PA2-Personal protocol.
- The shields are not working right now for technical reasons. (140+ hours spend programming and no luck)

14

# HC-SR04 Ultrasound Sensor:

- This model was chosen because it was designed to be used with the Arduino Microcontrollers and has stock supported code. The code does need to be for a 4 wire sensor.


15

# Terminal Strips:

- We are using terminal strips to distribute the power to other parts of the systems. The top strip has +12V and the bottom strip is a common ground. This allows for wiring changes, expansion, and upgrades for any future with ease.

16

# Battery:

- A common 12V, 7AH battery is used
- For longer protection a garden tractor battery can be used and instead of the small Sealed-Lead Acid battery that I had on hand.
- For the longer protection times a marine deep cycle 12 DC battery can also be used if extended periods of battery use is expected.

17

# Battery Charger:

- An automatic float and trickle charger a supplier will be used to maintain the battery.
- The tickle charger supplies a small amount of current at a voltage level higher that the battery voltage.
- The model used will automatically turn itself on and off as needed to keep the battery in peak condition.

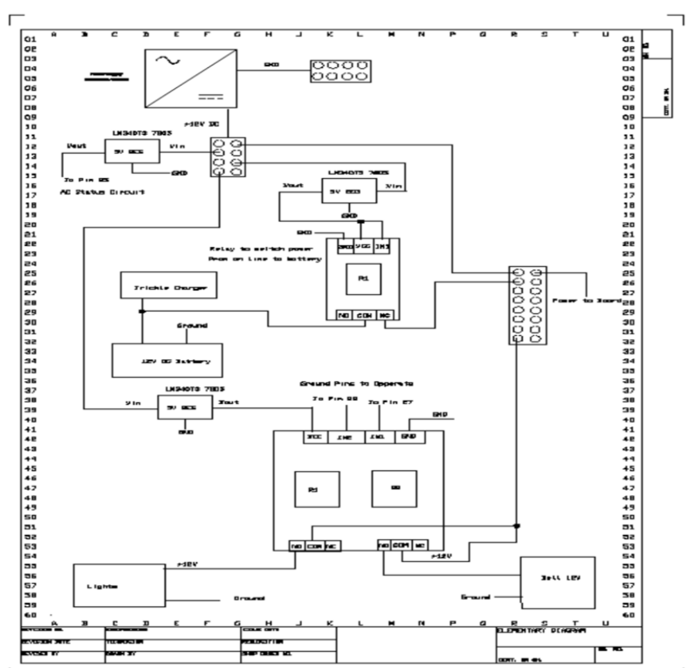
18

# LM340TS 7805:

- This is a positive 5 volts voltage regulator that takes between 7 and 20 Volts DC.
- I am using this chip to supply power to the relays.
- I am also sending a signal to the Arduino I/O to tell if the AC power is on.

19

# Wiring Diagram




20

# Programming States

- Below is the state diagram for the software
- This covers all the major “states” that the program will be in.

| Machine States | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| **Water Level** | **Status** | | **Has 5 Min Passed?** | **Water Level** | **AC Power Bit** | **Message** | |
| Low | Low Water level | | Yes | Low | Low | Power Off Pumps Off | |
| Midium | One Pump Running | | Yes | Midium | Low | Power Off, 2 Pumps On | |
| High | Both Pumps Running | | Yes | High | Low | Power Off, 1 Pumps On | |
| | | | Yes | Low | High | No Message | |
| **AC Power Bit** | **Status** | | Yes | Midium | High | 1 Pump On | |
| Low | AC Power Off | | Yes | High | High | 2 Pumps On | |
| High | AC Power On | | No | xxxxx | xxxx | No Message | |

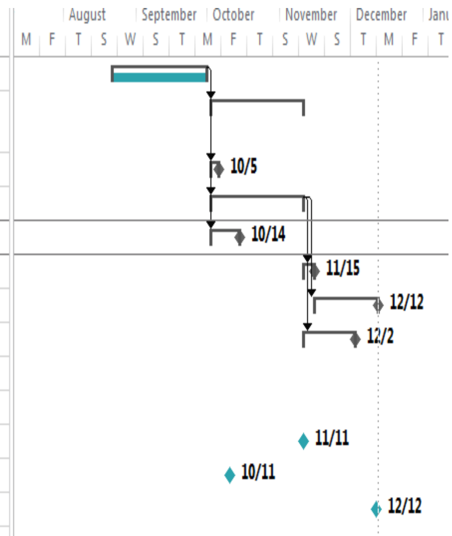


# Schedule

- On the following slides are the proposed schedule
- In reality programming was tackled first and the mile stones are a more accurate way of measuring progresses
- Over 40 hours spent on the WI-FI Shield
- Over 140 hours spent of the Ethernet Shield

# Schedule

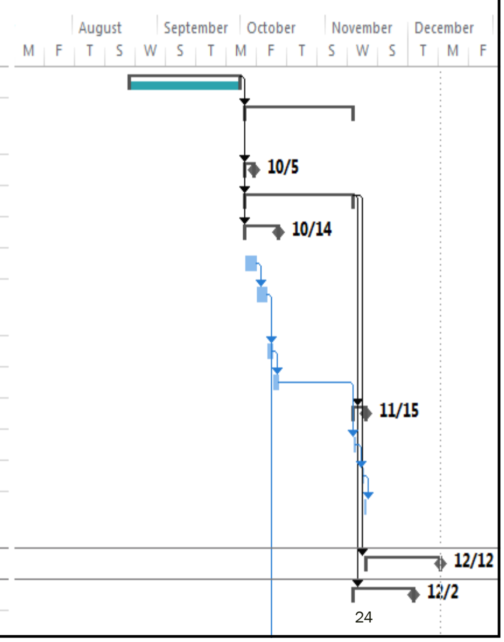
| Task Mode | Task Name | Duration | Start | Finish | Predecessors | Resource Names |
|---|---|---|---|---|---|---|
| | Prototype | 30 days | Mon 8/22/1 | Fri 9/30/16 | | |
| | Circuit Design Schematic Research | 29 days | Mon 10/3/16 | Thu 11/10/16 | 1 | |
| | Test Stand | 3 days | Mon 10/3/1 | Wed 10/5/1 | 1 | |
| | Arduino Programing | 29 days | Mon 10/3/1 | Thu 11/10/1 | 1 | |
| | Integration | 10 days | Mon 10/3/1 | Fri 10/14/16 | 1 | |
| | Testing | 3 days | Fri 11/11/16 | Tue 11/15/1 | 13 | |
| | Project Report | 19 days | Wed 11/16/ | Mon 12/12/1 | 13 | |
| | Presentation | 16 days | Fri 11/11/16 | Fri 12/2/16 | 13 | |
| | | | | | | |
| | Milestones | | | | | |
| | Design Finished | 0 days | Fri 11/11/16 | Fri 11/11/16 | 16 | |
| | Assembly Completed | 0 days | Tue 10/11/1 | Tue 10/11/1 | 19 | |
| | Final Presentation | 0 days | Mon 12/12/1 | Mon 12/12/1 | | |




23

# Schedule

| Task Name | Duration | Start | Finish | Predecessors | Resource Names |
|---|---|---|---|---|---|
| Prototype | 30 days | Mon 8/22/1 | Fri 9/30/16 | | |
| Circuit Design Schematic Research | 29 days | Mon 10/3/16 | Thu 11/10/16 | 1 | |
| Test Stand | 3 days | Mon 10/3/1 | Wed 10/5/1 | 1 | |
| Arduino Programing | 29 days | Mon 10/3/1 | Thu 11/10/1 | 1 | |
| Integration | 10 days | Mon 10/3/1 | Fri 10/14/16 | 1 | |
| Initial Assy | 4 days | Mon 10/3/1 | Thu 10/6/16 | 3 | |
| Hardware/software integration | 2 days | Fri 10/7/16 | Mon 10/10/16 | 18 | |
| Troubleshoot | 2 days | Tue 10/11/1 | Wed 10/12/1 | 19 | |
| Functional Testing | 2 days | Thu 10/13/1 | Fri 10/14/16 | 20 | |
| Testing | 3 days | Fri 11/11/16 | Tue 11/15/1 | 13 | |
| Water Level | 1 day | Fri 11/11/16 | Fri 11/11/16 | 21 | |
| Power Testing | 1 day | Mon 11/14/1 | Mon 11/14/1 | 23 | |
| Bluetooth | 1 day | Tue 11/15/16 | Tue 11/15/16 | 24 | |
| Project Report | 19 days | Wed 11/16/ | Mon 12/12/1 | 13 | |
| Presentation | 16 days | Fri 11/11/16 | Fri 12/2/16 | 13 | |




24

## Lessons Learned

- An email is easier to send than a text message so early on in the programming the decision was made to send an email instead.
- The Wi-Fi shield that I was planning on using would not work with the WAPA-2 Enterprise protocol so I made the decision to find an Ethernet shield and code the program with this in mind.
- This would have affected the final presentation on the project as they like to have a working circuit when you give the presentation.

25

## Lessons Learned

- Using predesigned components can save time in the design processes,
- For a final production piece a printed circuit board with surface mounted components is the way to go.
- This is very much a prototype design to prove proof of concept and to evaluate under long tern testing variables that cannot be measured or predicted in a lab environment.
- For example real word testing of the sensor with the sump pit to find how much interference moving water has on the readings, and ways to increase accuracy.

26

# Lessons Learned

- Obsolete components are hard to troubleshoot (Wi-Fi 101 new board)
- Connecting to the Internet is tricky
- Surface mount parts do go bad on Printed Circuit Boards
- Possibly burned the Ethernet Shield
- SMS Shield for texting is a possible work around



# Conclusion

- The circuit was build with off the shelf components to create a working prototype that will allow the design team to prove the concepts and allow for real world testing to be carried out.
- Future versions of the design can include Wi-Fi101, Bluetooth, Applications for smart phones, and user defined push buttons and status lights that are labeled.
- I believe many homeowners and small business owners would benefit from the peace of mind that the device would bring.

# Demonstration

29

# Questions?

30

# Work Cited


- https://www.arduino.cc/
- http://www.talonix.com/shop/category.aspx?catid=281


31