ZMHW Project: Map

Steak Electronics

11/2018 - 8/2019

Contents

1 Overview

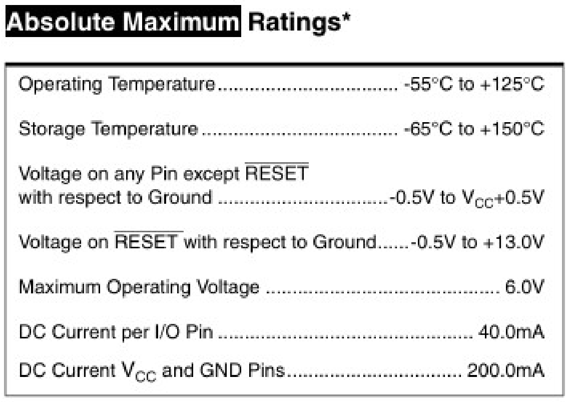
ZMHW Map is a device to generate a type of heat map, or map whereby you can visually see where on your property alarms are occuring. It does this by connecting to the Zoneminder server[1], and lighting LEDs on a PCB, with a CAD layout of your property on the PCB. By lighting certain LEDs that correspond to an alarm (with either motion, or better → hardware detection. See ZMHW Motion Detector [2]).

ZMHW Map is built as an Arduino sketch around the common Arduino Mega 2560, with a PCB layout (gerber files) that you can have fabbed[3]. Minimal experience with a soldering iron is required. The current layout allows for up to 45 alarms to be set. As not all alarms will be active at any given time, each LED is set to output at 5mA, and should be within the 100-200mA output pin limit of the atmega 2560.

[1]Via zmtrigger.pl

[2]https://git.steakelectronics.com/adminguy/ZMHWProject

[3]Fabbing can be done in the US within a span of about two weeks from purchase to delivery for a low price at MakerBright aka https://pcbs.io in Florida or OSH Park, https://oshpark.com based in Oregon. Users can also use overseas PCB services - see https://pcbshopper.com



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA

Figure 1: Atmega2560 Maximums

2 PCB Revision 1

Upon Building the first rev, I noticed a few things that I've found important with the other shields I've been making with the ENC28J60. First off, I want the ENC to be upside down, so that it fits snug between the Mega USB and Barrel plug. The ENC module is sold with male pin headers (not female) so it's a matter of placing it and soldering. However, I didn't realize how well it fit until I made the first shield of ZMHW Motion Sensor, so I hadn't yet flipped the pins. Rev2 will have the ENC upside down. The 5x2 pins are enough to hold it in place. Ideally, a custom enclosure would add additional support.

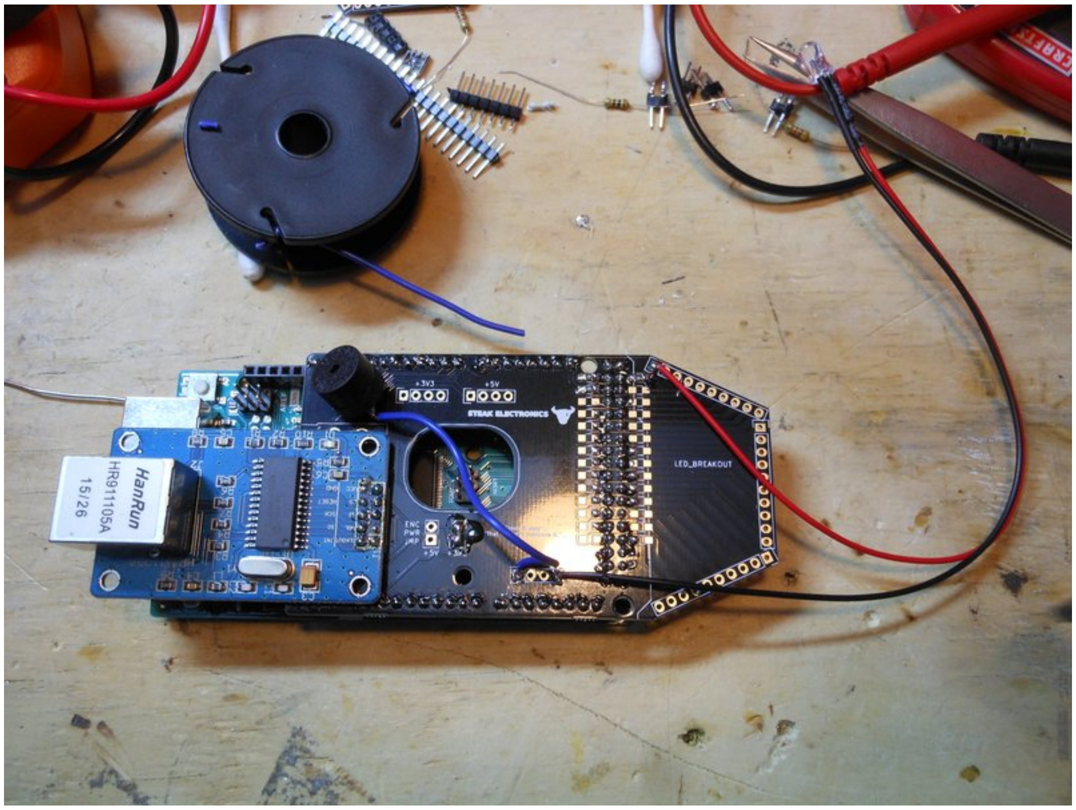


Figure 2: Board assembled with ENC, speaker, and pre-wired LEDs

2.1 Troubleshooting the board:

In this type of shield (NOTE that this won't be required in later revisions. This is one of the reasons, I flipped the ENC), where you must cut off the old ENC 0.1" headers, there is a trick to getting them off without damaging the module. You must cut each pair of headers at the plastic part (don't cut the actual pins) with a pair of flush cut pliers, then desolder each pair off. I accidentally pulled a pad when doing this, and found the error with a microscope.

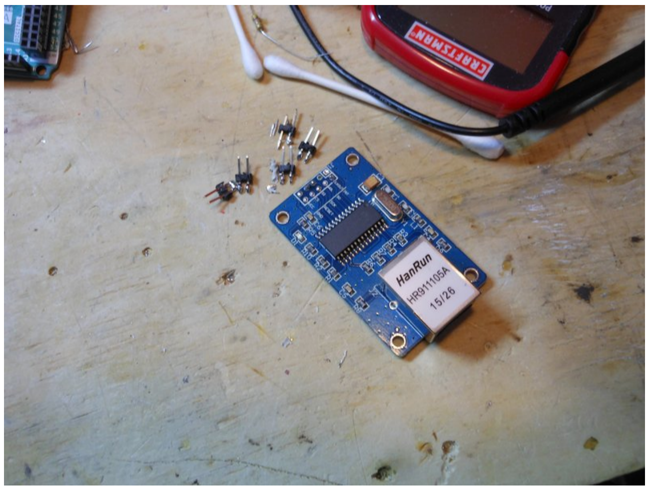


Figure 3: Cut plastic part off in pairs, then desolder. OR, design your board so you don't have to remove the pin headers!

Figure 4: If you are going to be soldering high gauge magnet wire, you should use a microscope. It makes the process much more enjoyable, and efficient.

2.2 Build Considerations

After I built this up, I realized that it was quite laborious to do soldering for the LEDs. The original idea was to buy a canvas print with an image of the property, and put LEDs through the canvas to alight where alarms were found. However, that would mean I might have to soldering wires for 30 LEDs (if you have 30 monitors), which is a slow, slow process. Instead, since I have already made PCBs of fairly large sizes, I will put the business property map on a PCB as the silk screen layer, and connect the board directly to the shield. I won't use a cable, as cables are an additional BOM item, and setup required, which is mentioned on the Amp Hour podcast (270 or so).

As for the option of using a shield with 30 wired LEDs vs. a large PCB, I've weighed the options of each. While the 30 wired LEDs may be a 'better' solution, the labour involved outweighs the benefits, and instead a PCB is the more practical choice. I can make large PCBs and solder LEDs on a board much quicker than I can make 30 wired LEDs. I intend to make a type of picture frame for the board, and mount it behind glass, so dust doesn't collect on it. This means each property requires its own PCB, but the building PCB is a simple one to make.

3 PCB - Revision 2

PCB Revision 2 was made, and while it works there were some caveats. First, I accidentally made my business property maps to start at the opposite end of the 50 Pin header than the main board. Oops. Second, the ENC pins were not placed correctly. Otherwise, I was able to hack around and get 1 prototype working, but I'll go to revision 3 to eliminate these issues. I also tested a PCB that was white, and a black PCB. While I've only actually lit lights on the white PCB, I think I prefer the black for this application.

On the plus side, the software code, which I had already made, is working without issue (in my short time testing) and overall, I am pleased with how the map works. It's a tool, and it gets the job done. You can quickly get an overview of where people are on your property. If you were to try to do this without the map, it would necessitate reviewing each camera stream, which means perhaps, 10-30 seconds. The map is more efficient. This is a new avenue of Zoneminder that isn't addressed currently. In addition, the random nature of the alarms going off, means that the map is always lighting up a new pattern of lights.

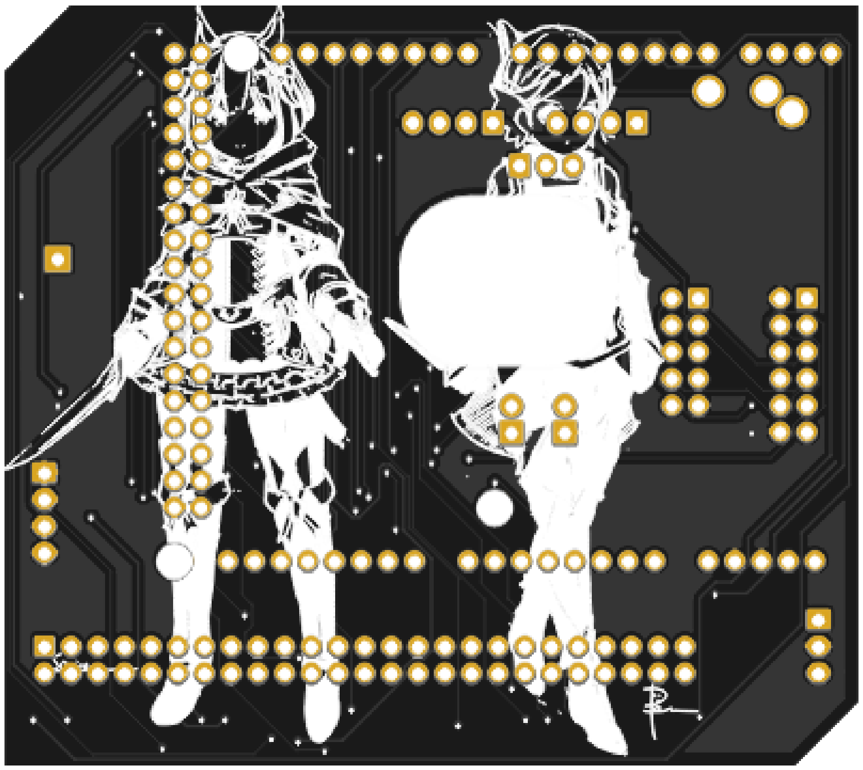


Figure 5: Revision 3 Back

4 PCB - Revision 3 Layout

On this PCB revision I did the following things:

- Flipped IO to start at the right side, instead of the left
- Fixed error with ENC28J60 Pins
- Added screw terminal for external RESET button
- Added picture to back

One thing I would like to do for the next revision: When I shortened the length of the MEGA footprint, I removed some pins, and RESET is one of the pins removed. I'd like to reinstate it. I'd also like to remove the silkscreen outline of the mega on the footprint.

4.1 PCB - Revision 3 Build

This revision works 100%. That is good. The bad thing is: assembly takes too long. I'm going to remake this with a led matrix, to avoid the need for 50 pins outside of the mega.

4.2 RGB Code, and Future Plans

I was able to build a mount out of a wooden board, and by sanding, and finishing with Urethane, I have a protected surface for the board.

I added RGB code, so that an LED can give a visual indication of the status. I found the RGB led shows program routines that you might not be aware were happening, as it can quickly indicate points in the code, well.

Future plans, are to use a smaller board, with an LED matrix, and an Arduino Uno instead of a Mega. With LED Matrix, I should be able to use the Uno, or even the Nano. There is no other need for all the pins. I don't think the telnet needs the SRAM of the Mega but I should double-check. If I make a board compatible with the uno, I can bring it to the Mega if necessary (compatible shield). I also plan to have the board behind the piece (or pieces) of wood, so only the map is visible. A connecter will be connected to the back of the map via a routed hole on the wood. Should look quite professional. I will make a new repo for this revision, to not obliterate the older code.

4.3 Mounting the boards to look Professional

The first time I made a board, I put the micro on the outside of the oak board. I've decided to instead hide the micro behind the board, and show only the map on the front. I will use a connector for this, and route a hole in the wood board so the micro is not visible. I'll connect to the back of the board.

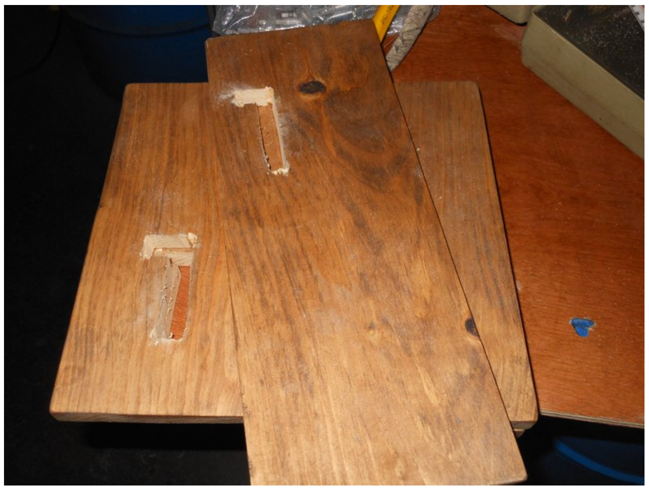


Figure 6: Instead, we will hide the micro behind the oak board, for a more professional look

I've also decided to stain the wood, before finishing. Stain makes it look a bit older and more rustic. See the Pictures.

5 480x320 TFT LCD Test

Some notes are in the readme.

To change brightness, contrast, etc... there are registers in the ILI9486 data sheet. the command in the MCUFriend library is readreg32. An example is given in the readID command of MCUFriend_kbv.cpp.

```
ret = readReg32(0xD3);      //for ILI9488, 9486, 9340, 9341
   msb = ret >> 8;
   if (msb == 0x93 || msb == 0x94 || msb == 0x98 || msb == 0x77 || msb == 0x16)
       return ret;             //0x9488, 9486, 9340, 9341, 7796
```

In the datasheet, d3h (hex D3) is the register that can be read, with ID info. Brightness is a register for writing, and one for reading.

Write registers is e.g.

```
WriteCmdData(0xB0, 0x0000);
```

I assume it is register,data. There are some register init data sort of examples in: extras/unused/mcufriend_special_2.h: of MCUfriend library. But to test the register peek/poke access, if I write

```
tft.WriteCmdData(0x51,0x55); //*Needs tft. to access MCUFriend commands
```

That should lower brightness. So use that in a loop. At first try, it didn't work. I'm trying to read a register, but it complains, readReg32 is protected, and error within this context. Bull. So I added this command to the bottom of the sketch.

```
uint32_t readReg32(uint16_t reg)
{
    uint16_t h = tft.readReg(reg, 0);
    uint16_t l = tft.readReg(reg, 1);
    return ((uint32_t) h << 16) | (l);
}
```

Maybe I should get myself a computer that allows writing and reading registers. Seems the Arduino can't do this. Documentation for this is abysmal. Poor documentation is job security. Perhaps I have to use the Adafruit commands in this case. i doubt it, though. Hardware support is in MCUFriend. Seems that OLEDs aren't supposed to adjust brightness, says adafruit forums. I hope that's not true. I gave up on this. I was unable to get the screen to print out text. Not letters, but I want it to passthrough sentences from serial. This way, whatever I send to it via UART displays on screen. I tried the serial.event tutorial (copied 1:1) from the official ide docs, but it did nothing...

5.1 The deal with brightness on this ILI9486

I found a discussion on this thankfully. It looks like there is 0 way to adjust brightness for these TFT screens. Here's what the comment says on https://github.com/prenticedavid/MCUFRIEND_kbv/issues/25

```
Both ILI9486 and ILI9481 are MIPI controllers.
You should use pushCommand() rather than WriteCmdData().
e.g. uint8_t val = 0x2C; tft.pushCommand(0x53, &val, 1);

Neither of the displays in your photos have access to the backlight. It is permanentl
There is little point in playing with reg(0x53) unless you have control of the backli

Just accept that you always lose ~200mA to a backlight whether the TFT is displaying
If you are battery powered, you will just have to go to the shops every few hours.

David.
```

I have no comments.

```
extras/unused/ILI9341_regValues.txt:        init_table(ILI9486_regValues, sizeof(ILI9
```

seems notable. also notable:

```
https://forums.adafruit.com/viewtopic.php?f=47&t=63229&p=320378&hilit=0xef+ili9341#p3
Seems tft's are corrupt. People shouldn't use bullshit.
```

Search for this code,

```
    case 0x9486:
        _lcd_capable = AUTO_READINC | MIPI_DCS_REV1 | MV_AXIS; //Red 3.5", Blue 3.5"
//        _lcd_capable = AUTO_READINC | MIPI_DCS_REV1 | MV_AXIS | REV_SCREEN; //old R
        static const uint8_t ILI9486_regValues[] PROGMEM = {
            0xC0, 2, 0x0d, 0x0d,        //Power Control 1 [0E 0E]
            0xC1, 2, 0x43, 0x00,        //Power Control 2 [43 00]
            0xC2, 1, 0x00,      //Power Control 3 [33]
            0xC5, 4, 0x00, 0x48, 0x00, 0x48,    //VCOM  Control 1 [00 40 00 40]
            0xB4, 1, 0x00,      //Inversion Control [00]
            0xB6, 3, 0x02, 0x02, 0x3B,  // Display Function Control [02 02 3B]
```

This is in MCUFriend_kbv.cpp. Let's change this to what the user had. link:

```
https://github.com/prenticedavid/MCUFRIEND_kbv/issues/31
 https://www.ebay.com/itm/240x320-2-4-SPI-TFT-LCD-Touch-Panel-Serial-Port-Module-with
```

Seems this might be a better screen, as it has a LED pin, to adjust brightness.

5.2 External Links

https://www.traintrackr.io/ - Here's a similar project. They made the same mistake I made - putting the leds on the front. I won't be doing that for rev 3.