Ayden Dauenhauer

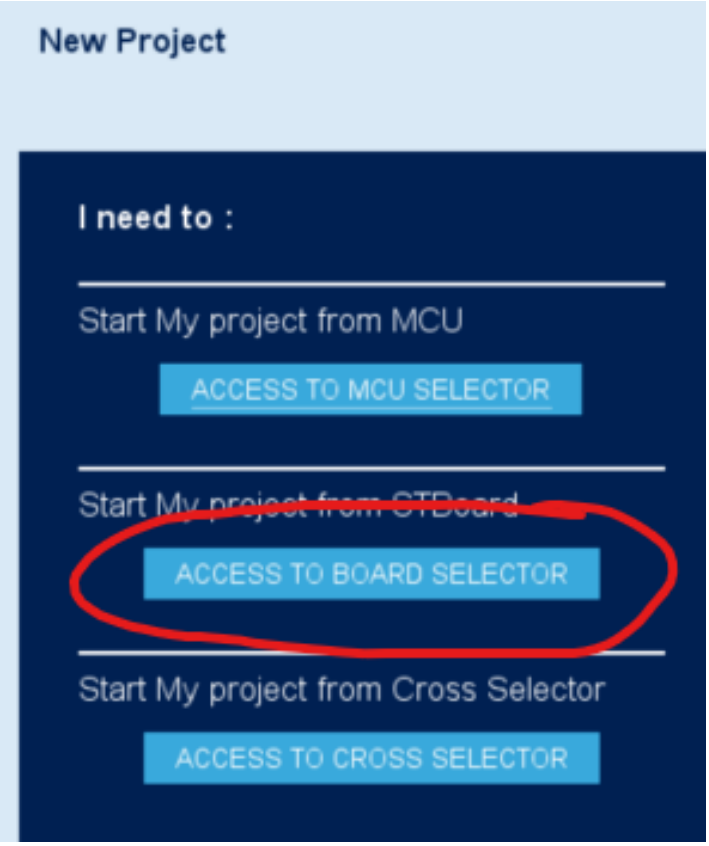
Prof. Wolfe

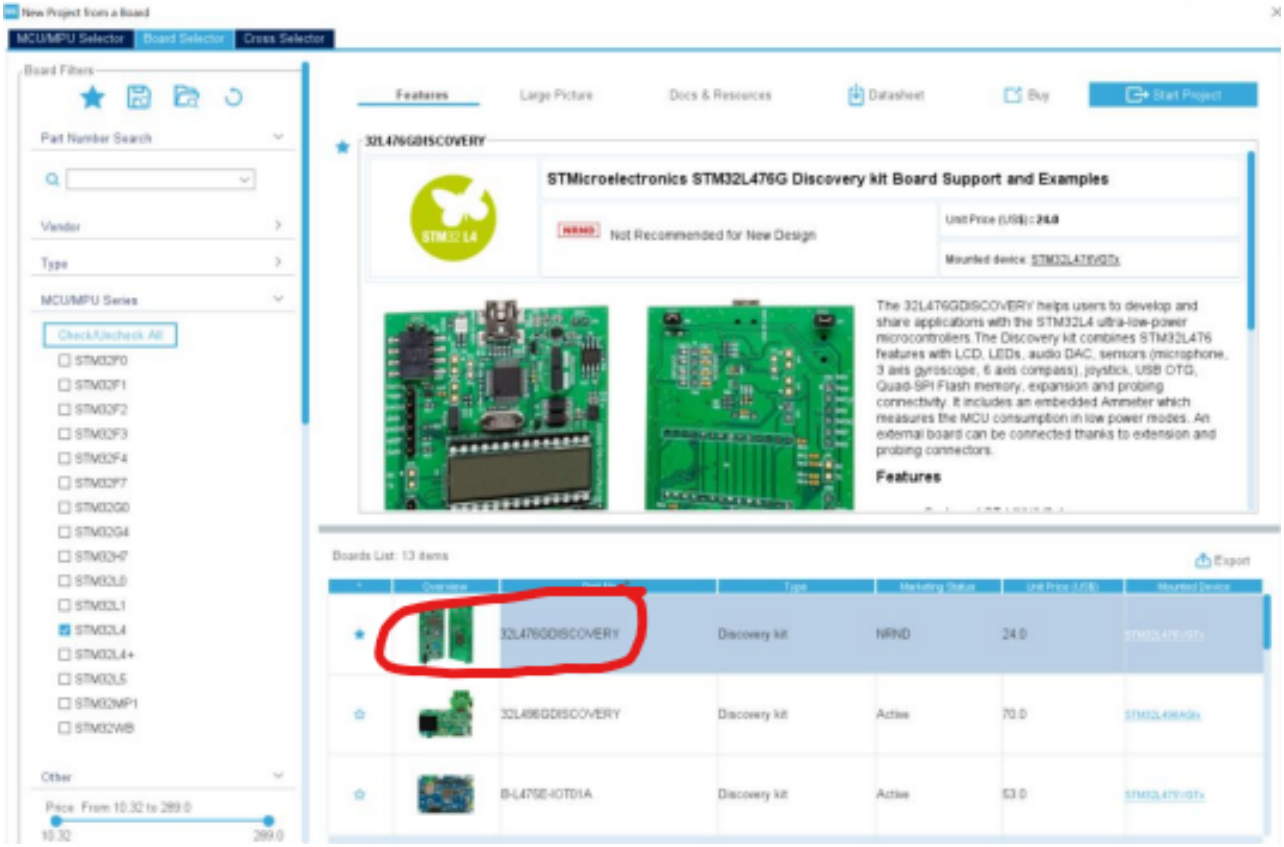
ELEN 121L Tuesday 2:15 p.m.

8 June 2024

Lab 1 – Writing and debugging some simple C programs

- Make sure the “Updater Settings” has the Z drive in the “Repository Folder”. This sometimes randomly changes!
- In CubeMX select “ACCESS TO BOARD SELECTOR”, then type STM32L476G and click on the discovery board.

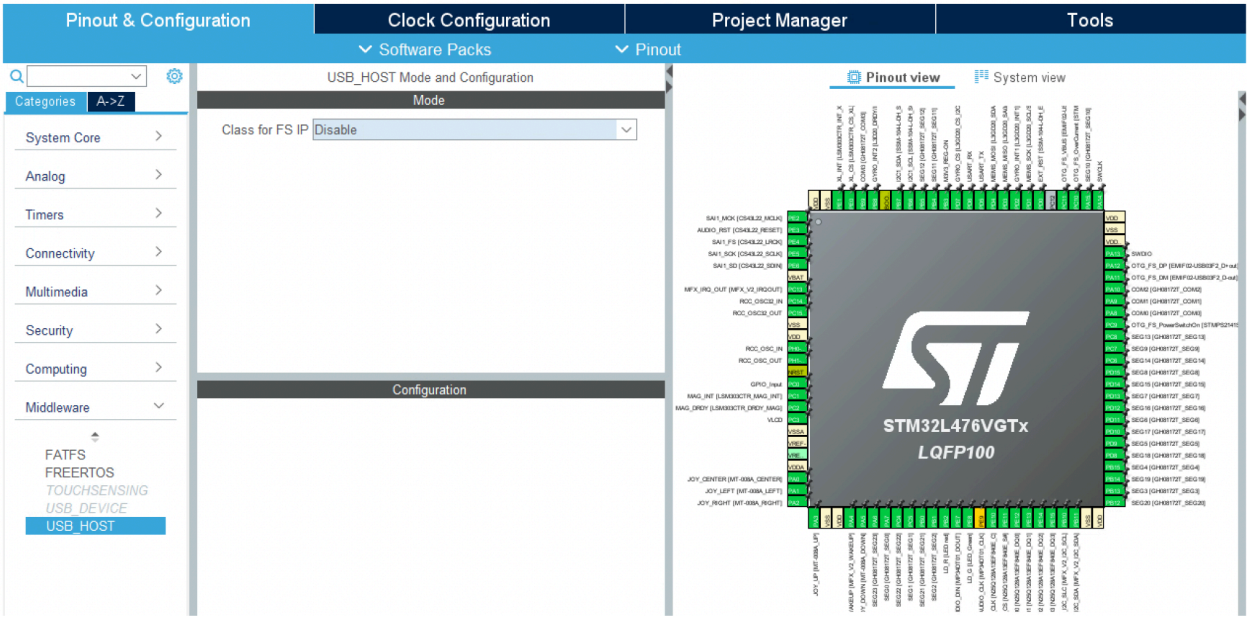




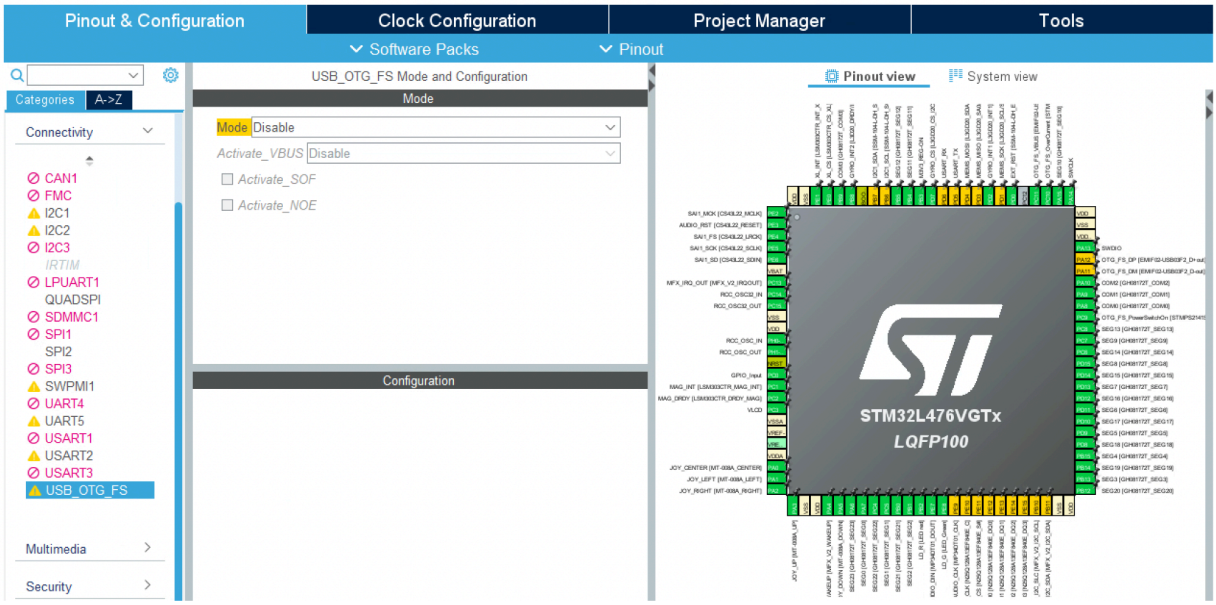
- GPIOB = Pin 2
- GPIOE = Pin 8

Lab 2 – Writing and debugging in the STM32cubeMX environment

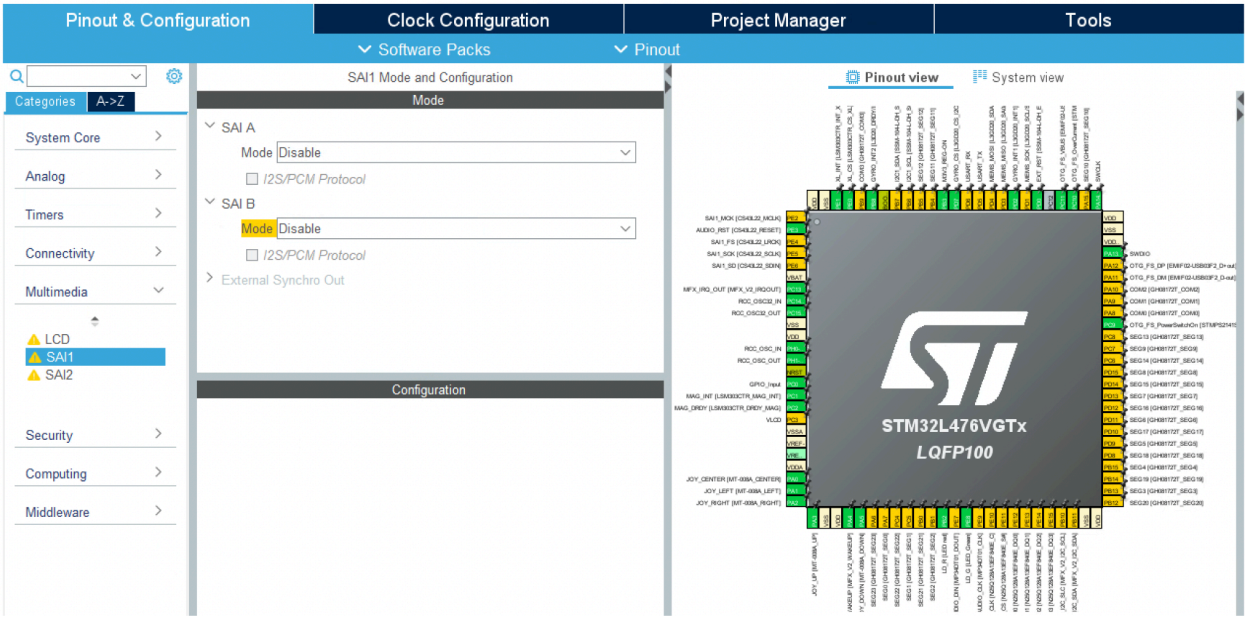
- Disabling Stuff on Chip
    - Disable USB: Middleware tab -> USB-HOST



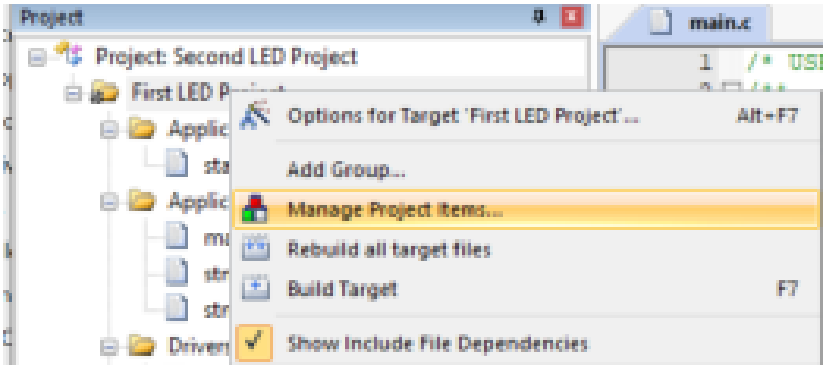
    - Disabling Others
        - Connectivity tab -> QuadSPI, I2C1, I2C2, SPI2, USART2, and USB_OTG_FS (Yellow warning is fine)



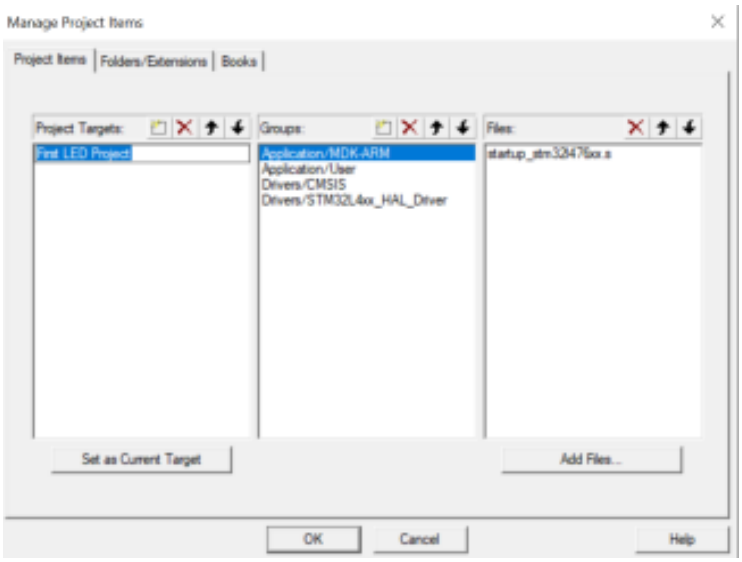
- Multimedia tab -> LCD and SAI1 (Disabling SAI A disables SAI B)



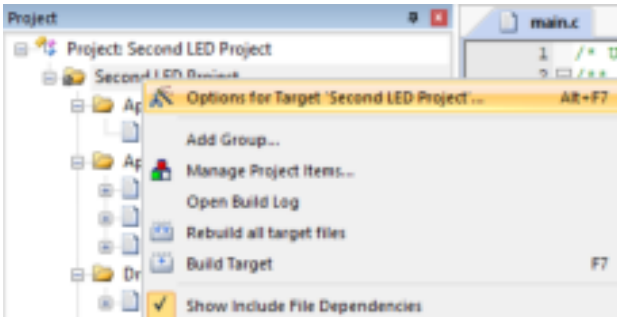
- USE METHOD 2 TO COPY PROJECTS!!
  - Copy-Paste project folder + rename
  - Rename .ioc
  - Delete .mxproject
  - Go into MDK-ARM and delete the 3 folders
  - Rename 3 Keil files
  - Create new folder with same name as project
  - Open Kiel, right click Project Folder and click “Manage Project Items”



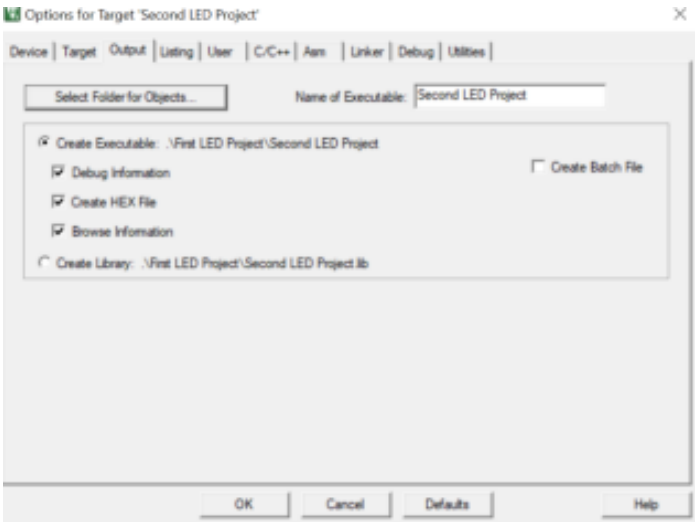
- Double click on the left and rename



- Right click Project Folder and click “Options for Target …”



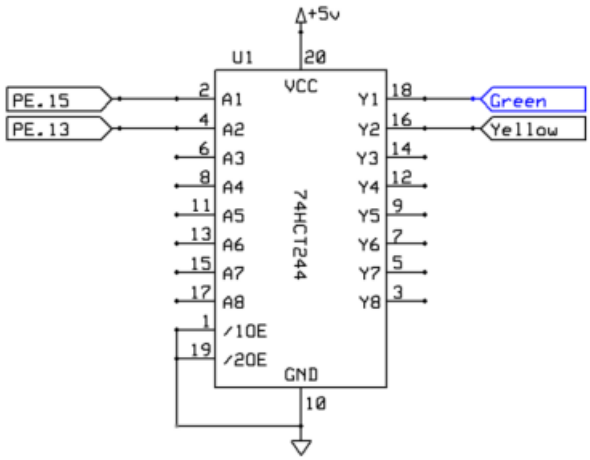
- Go to “Output” tab and rename the “Name of Executable”

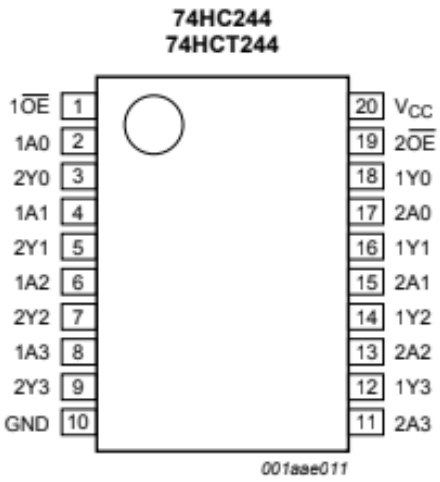


- Click “Select Folder for Objects” and select the project folder in the MDK-ARM folder

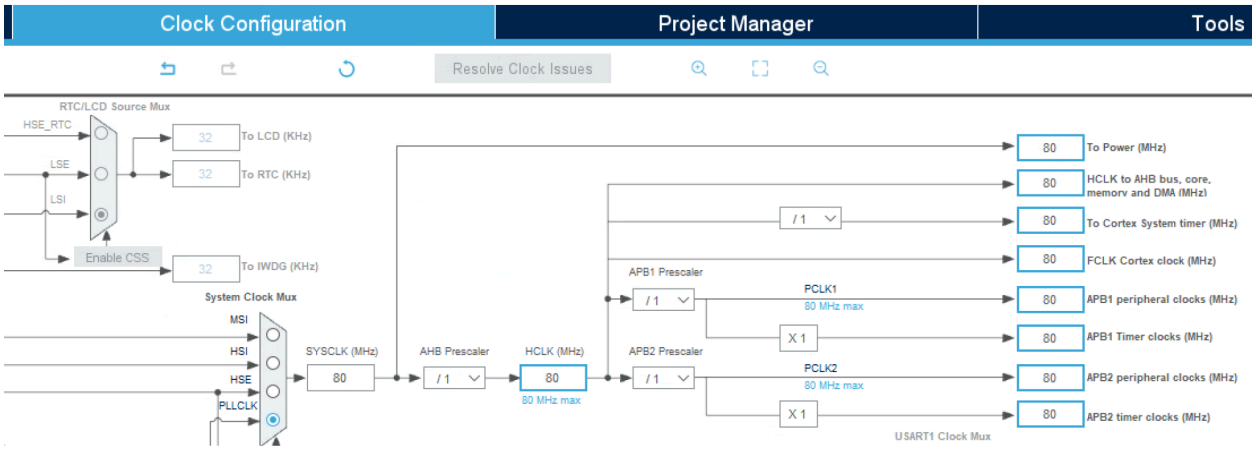
Lab 3 – Using the LED Strip

- Connecting board
  - Connect yellow, green, and black wires to the matching colors on the LED strip, leave red unconnected
  - Connect the black wire from the LED strip to ground

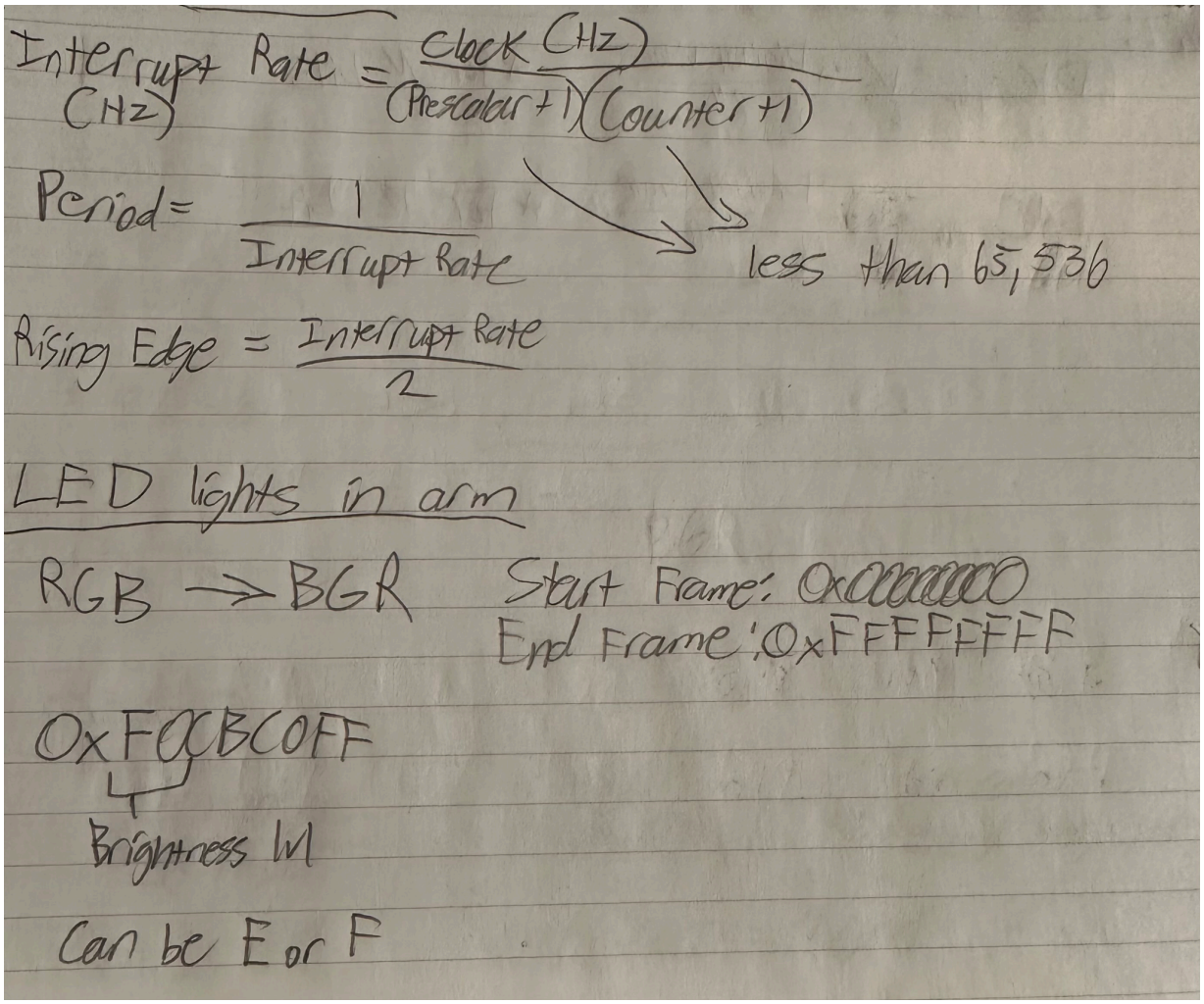




- Clock configuration
  - In the “Clock Configuration” change HCLK to 80 instead of default 20



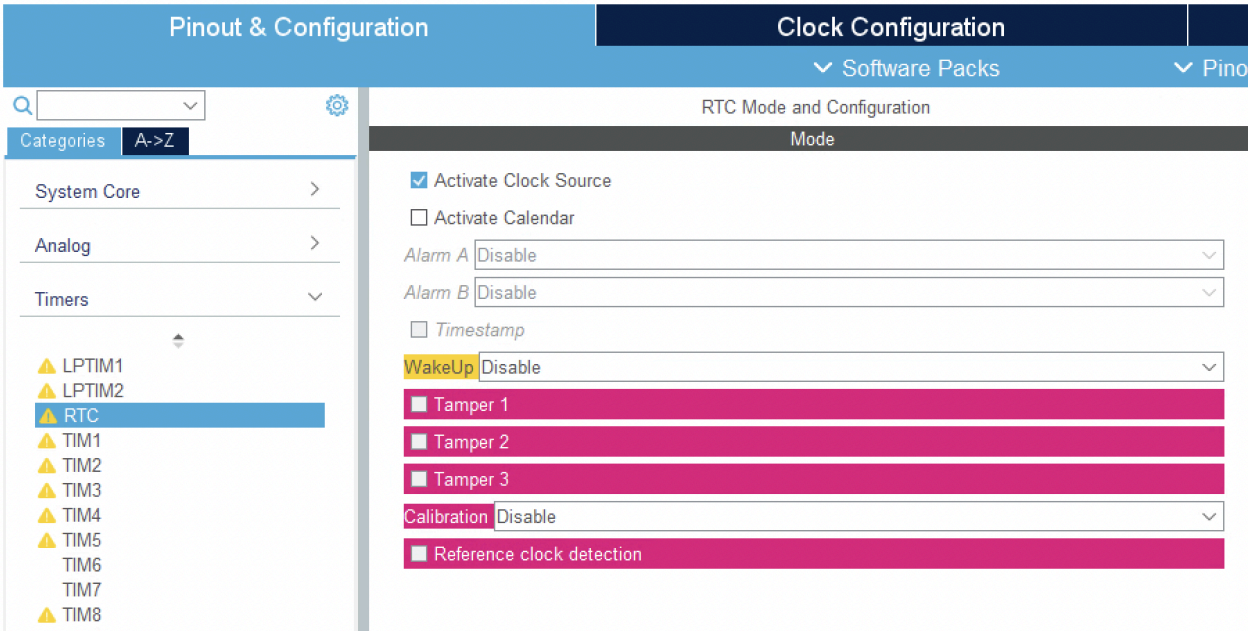
- Lights coded as BGR, include start and end frame



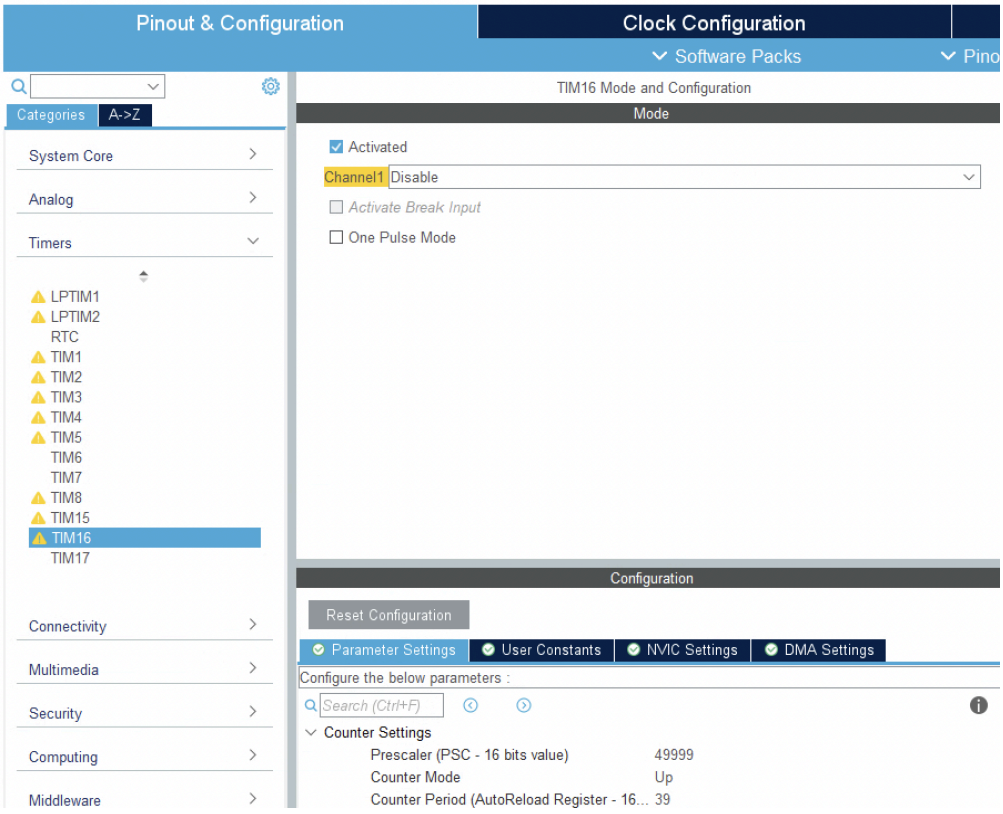
- Extra Credit: Use modulus and an array with color values

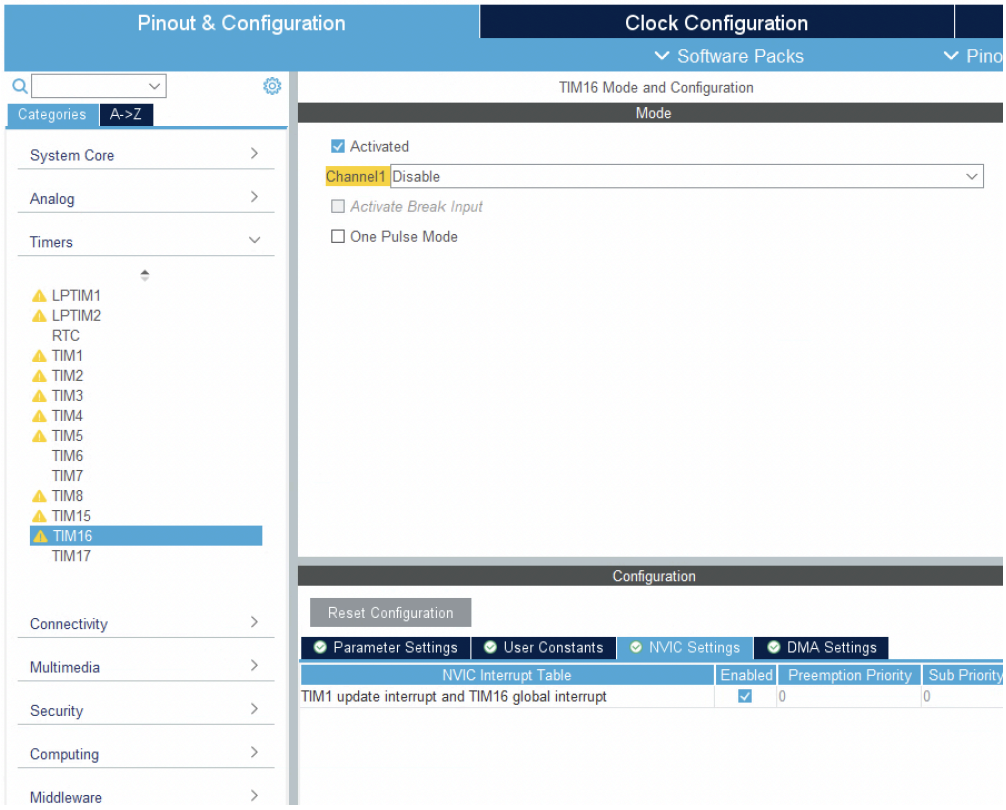
Lab 4 – Building an LCD Stopwatch

- DONT COPY OLD LABS, START FROM SCRATCH BECAUSE LCD WAS DISABLED!!
- Enable RTC clock in “Timers Section” by checking “Activate Clock Source”



- Enabling TIM16
    - Prescaler = 49,999
    - Counter Period = 39


Pinout & Configuration
Clock Configuration
Software Packs
Pino
Categories
A->Z
System Core
Analog
Timers
LPTIM1
LPTIM2
RTC
TIM1
TIM2
TIM3
TIM4
TIM5
TIM6
TIM7
TIM8
TIM15
TIM16
TIM17
Connectivity
Multimedia
Security
Computing
Middleware
TIM16 Mode and Configuration
Mode
Activated
Channel1 Disable
Activate Break Input
One Pulse Mode
Configuration
Reset Configuration
Parameter Settings
User Constants
NVIC Settings
DMA Settings
Configure the below parameters :
Search (Ctrl+F)
Counter Settings
Prescaler (PSC - 16 bits value) 49999
Counter Mode Up
Counter Period (AutoReload Register - 16... 39


Pinout & Configuration
Clock Configuration
Software Packs
Pino
Categories
A->Z
System Core
Analog
Timers
LPTIM1
LPTIM2
RTC
TIM1
TIM2
TIM3
TIM4
TIM5
TIM6
TIM7
TIM8
TIM15
TIM16
TIM17
Connectivity
Multimedia
Security
Computing
Middleware
TIM16 Mode and Configuration
Mode
Activated
Channel1 Disable
Activate Break Input
One Pulse Mode
Configuration
Reset Configuration
Parameter Settings
User Constants
NVIC Settings
DMA Settings
NVIC Interrupt Table
Enabled
Preemption Priority
Sub Priority
TIM1 update interrupt and TIM16 global interrupt
0
0

Lab 5 – Concurrency and Buffer Management

- Timer TIM3 configured to interrupt every 10ms (20MHz CPU clock)
    - Prescaler = 19,999
    - Counter Period = 9
    - Enable Interrupt in NVIC
- Timer TIM6 configured to interrupt every 48ms (20MHz CPU clock)
    - Prescaler = 19,999
    - Counter Period = 47
    - Enable Interrupt in NVIC
- Center joystick button configured to interrupt

Lab 6 – Consumer Infra-Red Remote Control Receiver

FEATURES

- Very low supply current
- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Supply voltage: 2.5 V to 5.5 V
- Improved immunity against ambient light
- Insensitive to supply voltage ripple and noise
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912

Pb Pb-free
e3
RoHS COMPLIANT
HALOGEN FREE
GREEN (5-2008)

LINKS TO ADDITIONAL RESOURCES

Product Page | 3D Models | Calculators | Marking | Packages

MECHANICAL DATA

Pinning for TSOP382.., TSOP384..:

1 = OUT, 2 = GND, 3 = V_S

- OUT = PD0, name the pin “IR_IN” in CubeMX, Vs = 3.3V (3V3 on board)
- Oriented based on the side with the semicircle
- Timer TIM7 configured to interrupt every 100µs (20MHz CPU clock)
    - Prescaler = 199
    - Counter Period = 9

- Enable Interrupt in NVIC

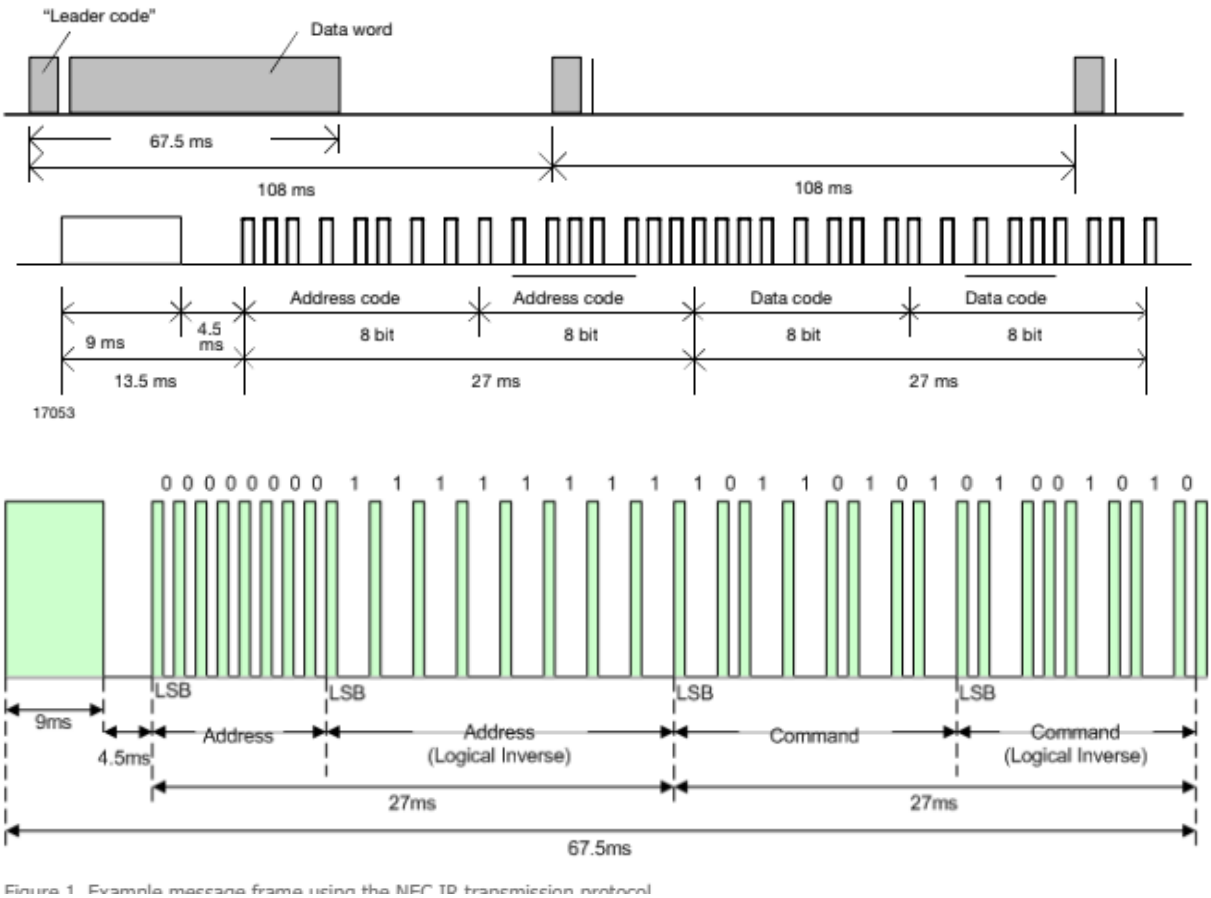


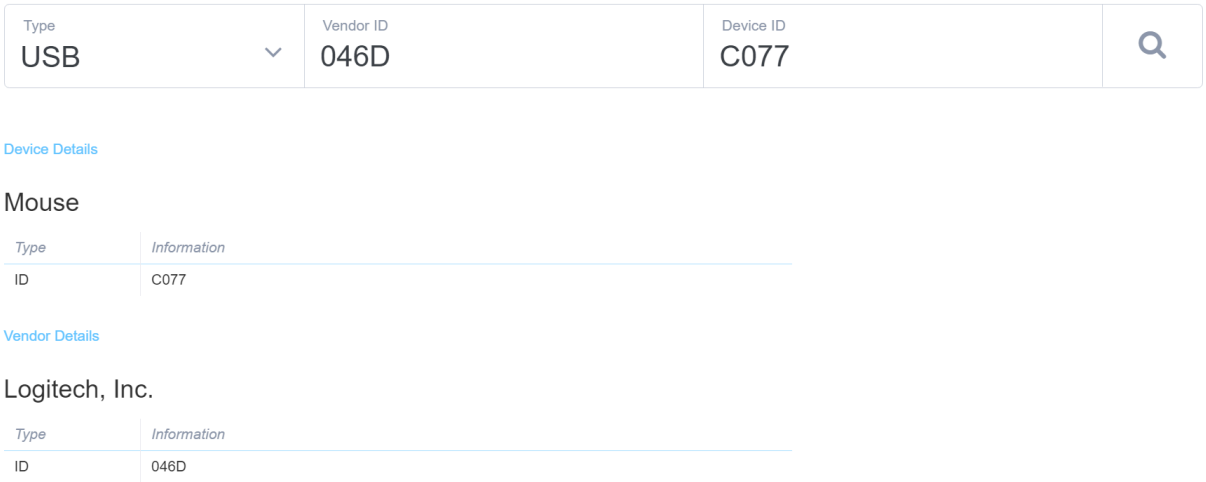
Figure 1. Example message frame using the NEC IR transmission protocol

- 27ms to transmit both the 16 bits for the address (address + inverse) and the 16 bits for the command (command + inverse). This comes from each of the 16 bit blocks ultimately containing eight '0's and eight '1's - giving (8 * 1.125ms) + (8 * 2.25ms).
- 67.5ms to fully transmit the message frame (discounting the final 562.5µs pulse burst that signifies the end of message).
- Bits are inverted, so either invert the bits again, or invert the ‘0’ and ‘1’ in logic of code
- IR_A = 0x00ff22dd
- IR_B = 0x00ff02fd
- IR_C = 0x00ffc23d
- IR_POWER = 0x00ff629d

- IR_UP = 0x00ff9867
- IR_DOWN = 0x00ff38c7
- IR_LEFT = 0x00ff30cf
- IR_RIGHT = 0x00ff7a85
- #IR_CIRCLE = 0x00ff18e7

Lab 7 – USB Mouse

- SPOTFRAC: Mouse sensitivity
- Vendor ID = 0x046D
- Product ID = 0xC077



- 16 Pipes

```
uint32_t                    Pipes[16];
```

- USBH_HID_SOFProcess called when initializing HID class

```
static USBH_StatusTypeDef USBH_HID_SOFProcess(USBH_HandleTypeDef *phost);
static void  USBH_HID_ParseHIDDesc(HID_DescTypeDef *desc, uint8_t *buf);

extern USBH_StatusTypeDef USBH_HID_MouseInit(USBH_HandleTypeDef *phost);
extern USBH_StatusTypeDef USBH_HID_KeybdInit(USBH_HandleTypeDef *phost);

USBH_ClassTypeDef  HID_Class =
{
  "HID",
  USB_HID_CLASS,
  USBH_HID_InterfaceInit,
  USBH_HID_InterfaceDeInit,
  USBH_HID_ClassRequest,
  USBH_HID_Process,
  USBH_HID_SOFProcess,
  NULL,
};
```

- USBH_UsrLog: creates a log of the device plugged in

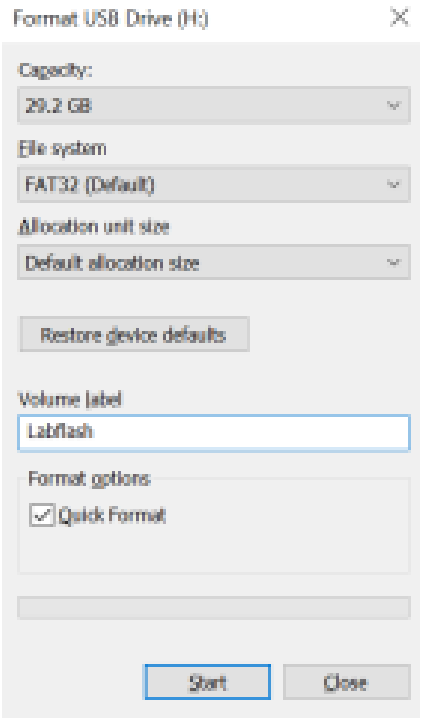
```
USBH_UsrLog("PID: %xh", phost->device.DevDesc.idProduct);
USBH_UsrLog("VID: %xh", phost->device.DevDesc.idVendor);
```

```
#define  USBH_UsrLog(...)   do { \
                              printf(__VA_ARGS__); \
                              printf("\n"); \
} while (0)
#else
#define USBH_UsrLog(...) do {} while (0)
#endif
```

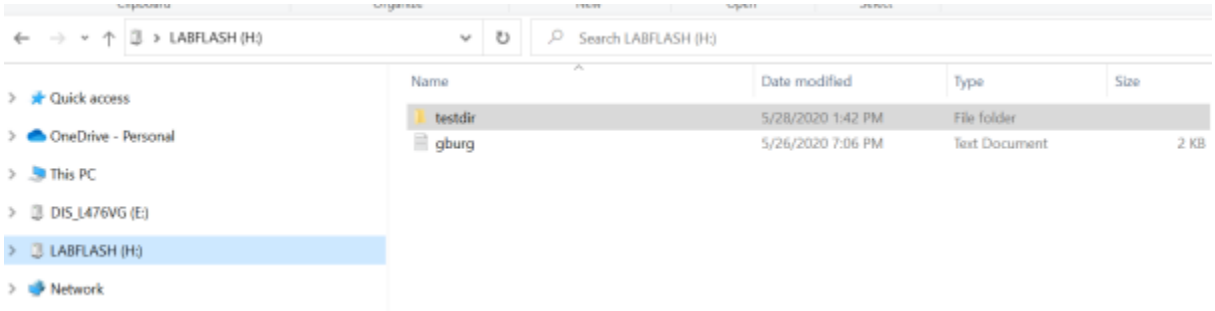
- fixData: mask which sign extends if the number is negative
- Left Click: mouse_info.buttons[0]
- Right Click : mouse_info.buttons[1]
- Middle Click: mouse_info.buttons[2]

Lab 8 – USB Flash Drive

- Connect flash drive to pc, open file explorer, right click the flash drive, select format



- Copy zip files into flash drive, ensure it looks like this, then eject

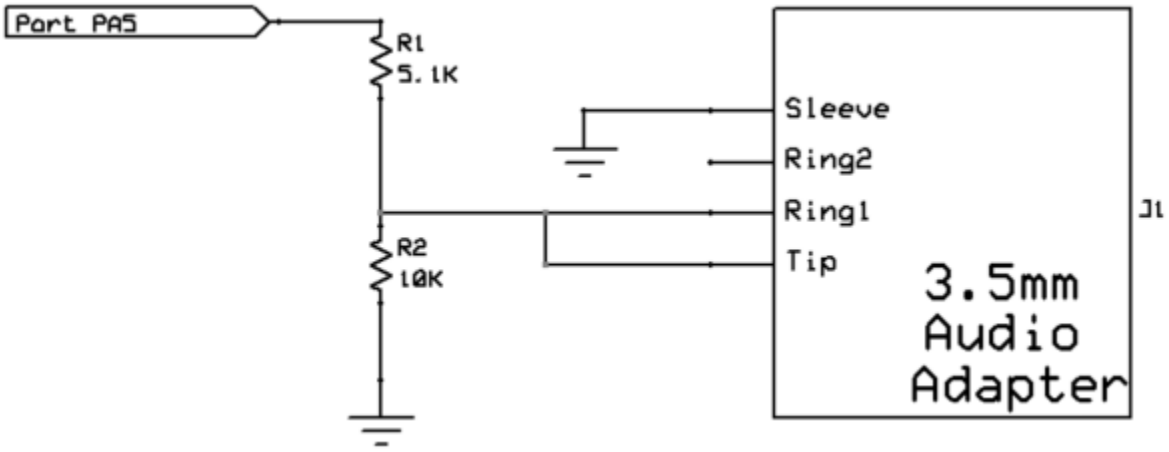


- FAT file lookup website: http://elm-chan.org/fsw/ff/00index_e.html
- Don't do part 2 from scratch, just download Lab8p2.zip
- Contents of the DIRLIST.TXT file after the program runs:

```
/DIRLIST.TXT
/SYSTEM~1/WPSETT~1.DAT
/SYSTEM~1/INDEXE~1
/GBURG.TXT
/TESTDIR/DIR1/F3.TXT
/TESTDIR/DIR1/DIR1A/F1.TXT
/TESTDIR/DIR1/DIR1A/F2.TXT
/TESTDIR/DIR2/F4.TXT
/TESTDIR/DIR2/F5.TXT
/TESTDIR/DIR2/F6.RTF
/TESTDIR/DIR3/PPT1~1.PPT
/TESTDIR/DIR4/DIRX~1/DIRY~1/DIR3/PPT1~1.PPT
/TESTDIR/DIR4/DIRX~1/DIRY~1/DIR1/F3.TXT
/TESTDIR/DIR4/DIRX~1/DIRY~1/DIR1/DIR1A/F1.TXT
/TESTDIR/DIR4/DIRX~1/DIRY~1/DIR1/DIR1A/F2.TXT
```

Lab 9 – USB Music Player

- Connect the speaker adapter to board



- Timer 6 interrupts 16,000 times per second (DAC1 Channel 2, outputs on Port A Pin 5)
- write_DAC1Ch2(uint16_t dacval, int volume) writes an unsigned 12-bit value to DAC
- Part 3: Copy lab 4 code, enable TIM16 and RTC in CubeMX
- Part 4: Copy lab 6 code, enable TIM7
- Timer TIM7 configured to interrupt every 100µs (80MHz CPU clock)
    - Prescaler = 799
    - Counter Period = 9
    - Enable Interrupt in NVIC