Worksheet #5 Task Screenshots & Pictures Jeremiah Webb

Task 1:

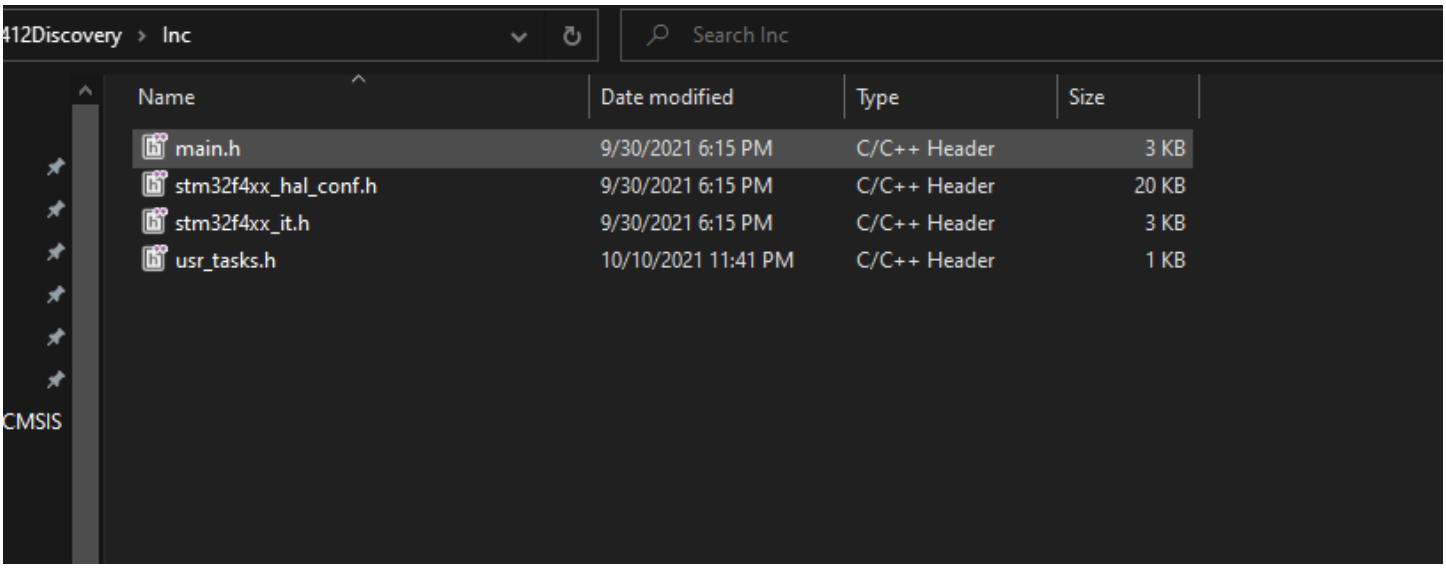
```
Build Output
Build started: Project: F412Discovery
*** Using Compiler 'V5.06 update 7 (build 960)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'F412Discovery'
compiling usr_tasks.c...
../Inc/usr_tasks.h(15): warning:  #1-D: last line of file ends without a newline
  #endif /* __USER_TASKS_H */
..\usr_src\usr_tasks.c: 1 warning, 0 errors
linking...
Program Size: Code=3250 RO-data=502 RW-data=16 ZI-data=1632
FromELF: creating hex file...
"C:\pjct_arm\build\F412Discovery\expl_014_template_for_HW_prjt.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed:  00:00:01
```

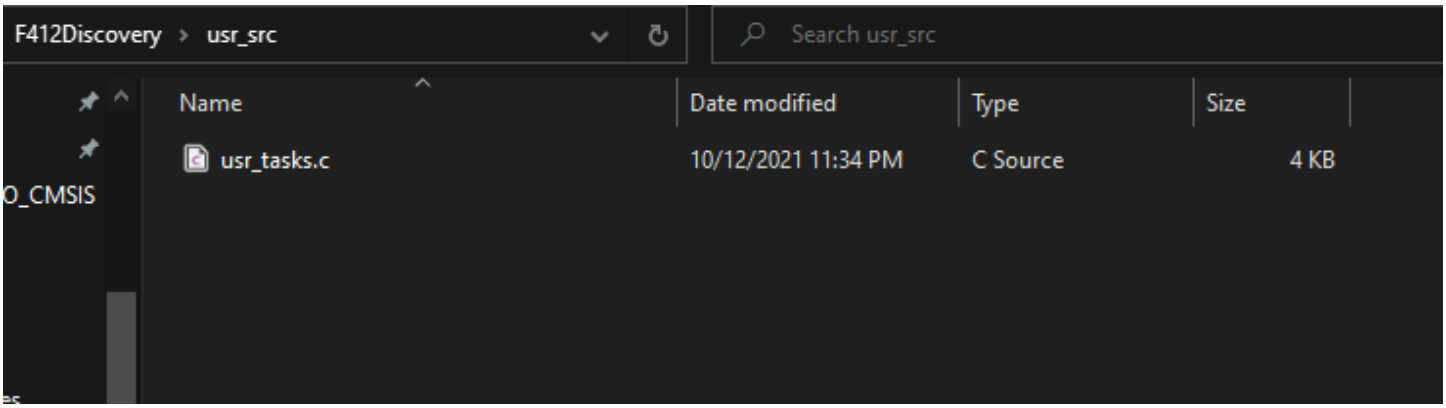
Task 2:

main.c | usr_tasks.c | usr_tasks.h

```
1  #ifndef __USER_TASKS_H
2  #define __USER_TASKS_H
3  #ifdef __cplusplus
4  extern "C" {
5  #endif
6  ////////////////////////////
7  #include "main.h"
8  #include <stdbool.h>
9  void USR_Task_RunLoop(void);
10 ////////////////////////////
11
12 #ifdef __cplusplus
13 }
14 #endif
15 #endif /* __USER_TASKS_H */
```

For some reason, when I put usr_tasks.h into the src folder, the program didn't work, so I put it in the inc folder.






Task 3.1:

main.c | usr_tasks.c | usr_tasks.h | stm32f4xx_hal_g

```
102 }
103
104 static void LD_R_Off(void) {
105   CMS_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, LI
106 }
107
108 static void LD_G_On(void) {
109   CMS_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, LI
110 }
111 static void LD_G_Off(void) {
112   CMS_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, LI
113 }
114
115 static void LD_R_Toggle(){
116
117 CMS_GPIO_TogglePin(LD_R_GPIO_Port, LD_R_Pin);
118 }
119
120 static void LD_G_Toggle(){
121   CMS_GPIO_TogglePin(LD_G_GPIO_Port, LD_G_Pin);
122 }
123
124 void USR_Task_RunLoop(void) {
125
126 if (JOY_L_Is_Pressed()) {
127
128     LD_R_On();
129     LD_G_Off();
130
131 }else if (JOY_R_Is_Pressed()) {
132
133   LD_G_On();
134   LD_R_Off();
135
136 }else {
137
138     LD_R_Toggle();
139     HAL_Delay(50);
140     LD_G_Toggle();
141     HAL_Delay(50);
142
143     }
144 }
```

```
11  // Prototype of static (private) functions
12  static bool JOY_L_Is_Pressed(void);
13  static bool JOY_R_Is_Pressed(void);
14  static void LD_R_Toggle(void);
15  static void LD_G_Toggle(void);
16  static void LD_R_On(void);
17  static void LD_R_Off(void);
18  static void LD_G_On(void);
19  static void LD_G_Off(void);
20
```

Task 3.2

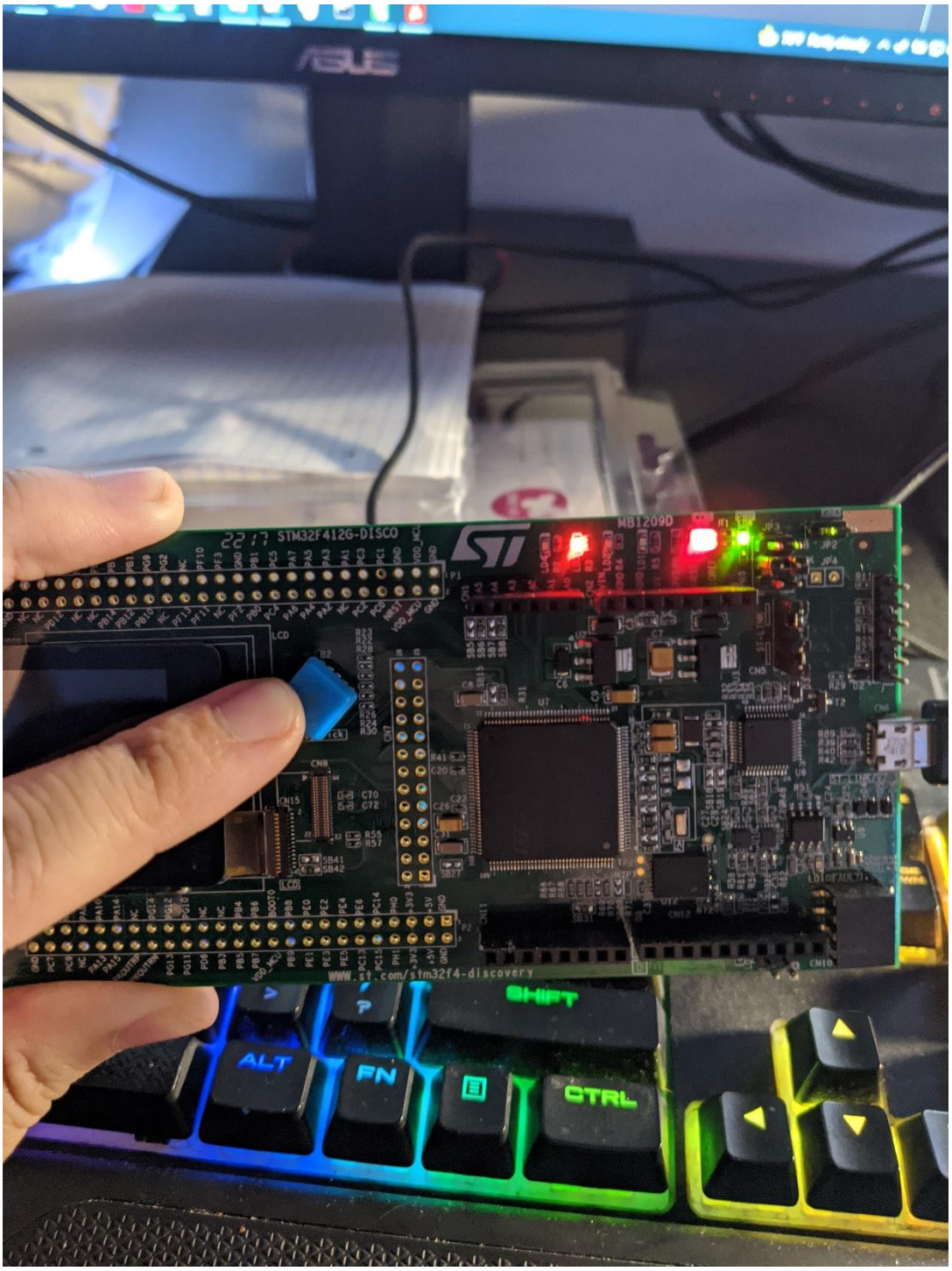
```
// Prototype of static (private) functions
static bool JOY_L_Is_Pressed(void);
static bool JOY_R_Is_Pressed(void);
static void LD_R_Toggle(void);
static void LD_G_Toggle(void);
static void LD_R_On(void);
static void LD_R_Off(void);
static void LD_G_On(void);
static void LD_G_Off(void);

/* original Functions I had
static bool JOY_L_Is_Pressed(void) {
return HAL_GPIO_ReadPin(JOY_L_GPIO_Port, JOY_L_Pin);
}
static bool JOY_R_Is_Pressed(void){

  return HAL_GPIO_ReadPin(JOY_R_GPIO_Port, JOY_R_Pin);
}
static void LD_R_On(void) {
  HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, LED_ON);
}

static void LD_R_Off(void) {
  HAL_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, LED_OFF);
}

static void LD_G_On(void) {
  HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, LED_ON);
}
static void LD_G_Off(void) {
  HAL_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, LED_OFF);
}

static void LD_R_Toggle(){

HAL_GPIO_TogglePin(LD_R_GPIO_Port, LD_R_Pin);
}

static void LD_G_Toggle(){
  HAL_GPIO_TogglePin(LD_G_GPIO_Port, LD_G_Pin);
}
*/
```

Task 3.3

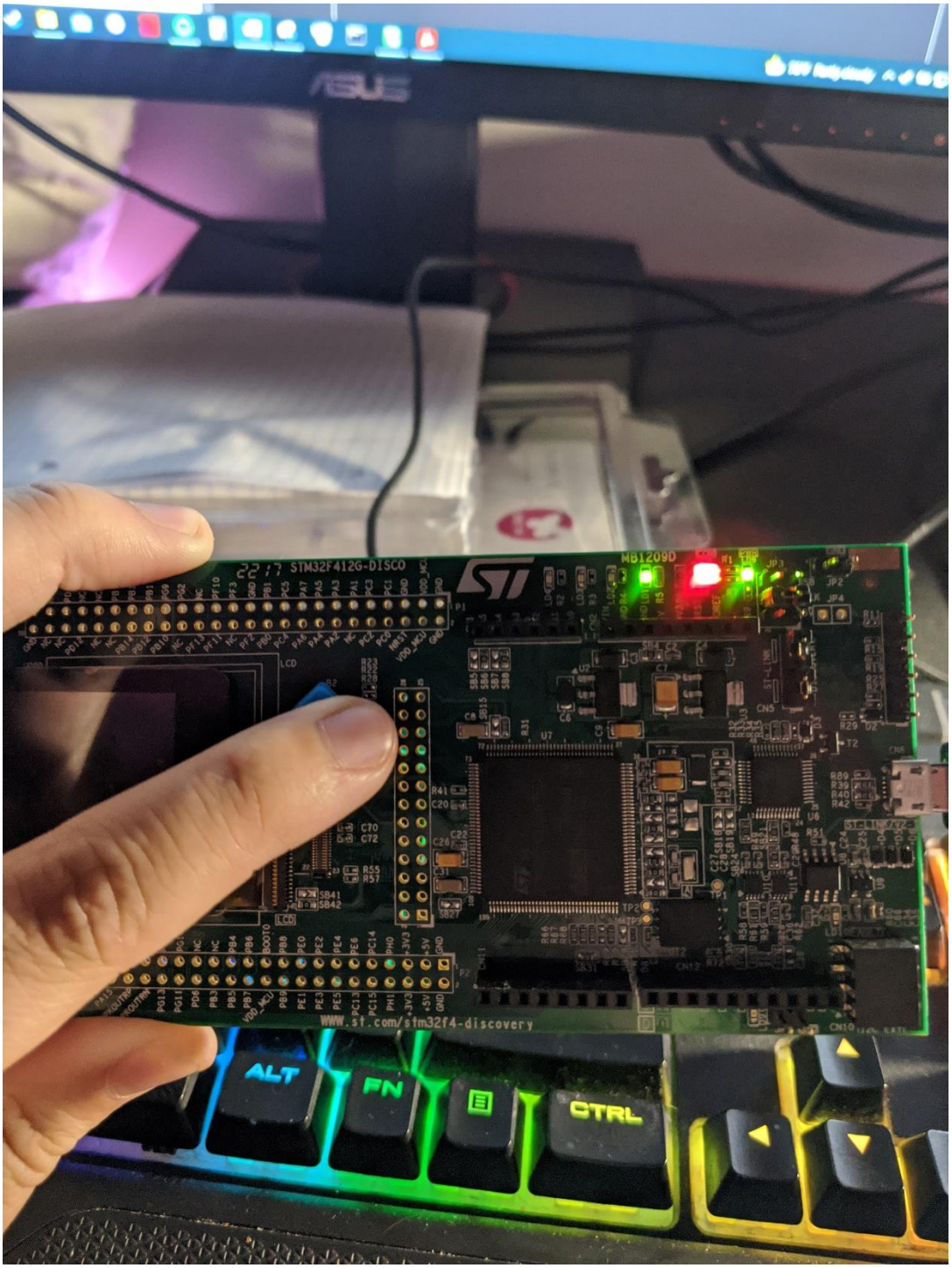
```
54  //CMSIS Operations
55  static void CMS_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState) {
56  if(PinState != GPIO_PIN_RESET) {
57    GPIOx->ODR |= (uint32_t)GPIO_Pin;
58  } else {
59    GPIOx->ODR &= ~((uint32_t)GPIO_Pin);
60  }
61  }
62  void CMS_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
63  {
64    assert_param(IS_GPIO_PIN(GPIO_Pin));
65    if ((GPIOx->ODR & GPIO_Pin) == GPIO_Pin)
66    {
67      GPIOx->BSRR = (uint32_t)GPIO_Pin << GPIO_NUMBER;
68    }
69    else
70    {
71      GPIOx->BSRR = GPIO_Pin;
72    }
73  }
74
75  GPIO_PinState CMS_GPIO_ReadPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
76  {
77    GPIO_PinState bitstatus;
78    assert_param(IS_GPIO_PIN(GPIO_Pin));
79
80    if((GPIOx->IDR & GPIO_Pin) != (uint32_t)GPIO_PIN_RESET)
81    {
82      bitstatus = GPIO_PIN_SET;
83    }
84    else
85    {
86      bitstatus = GPIO_PIN_RESET;
87    }
88    return bitstatus;
89  }
90
```

```
91  static bool JOY_L_Is_Pressed(void) {
92  return CMS_GPIO_ReadPin(JOY_L_GPIO_Port, JOY_L_Pin);
93  }
94  static bool JOY_R_Is_Pressed(void){
95
96    return CMS_GPIO_ReadPin(JOY_R_GPIO_Port, JOY_R_Pin);
97  }
98  static void LD_R_On(void) {
99    CMS_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, LED_ON);
100 }
101
102 static void LD_R_Off(void) {
103   CMS_GPIO_WritePin(LD_R_GPIO_Port, LD_R_Pin, LED_OFF);
104 }
105
106 static void LD_G_On(void) {
107   CMS_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, LED_ON);
108 }
109 static void LD_G_Off(void) {
110   CMS_GPIO_WritePin(LD_G_GPIO_Port, LD_G_Pin, LED_OFF);
111 }
112
113 static void LD_R_Toggle(){
114
115 CMS_GPIO_TogglePin(LD_R_GPIO_Port, LD_R_Pin);
116 }
117
118 static void LD_G_Toggle(){
119   CMS_GPIO_TogglePin(LD_G_GPIO_Port, LD_G_Pin);
120 }
121
```

Photos Showing Functionality:




Pressing Left, Turn on Red, turn off Green.




Pressing Right, Green is on, red is off.