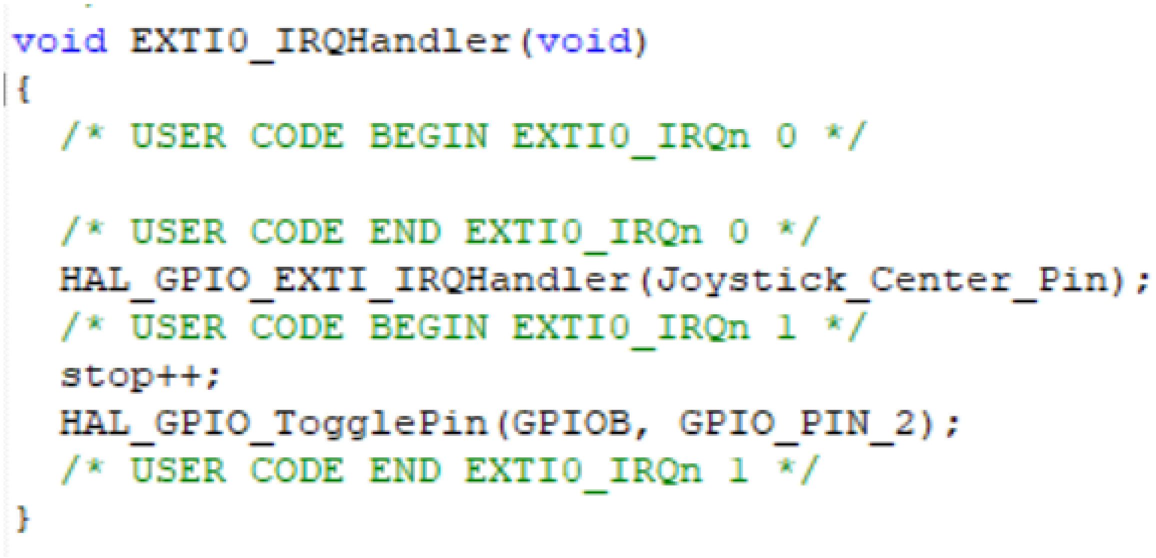
Alan Rieger and Ayden Dauenhauer

5/2/24

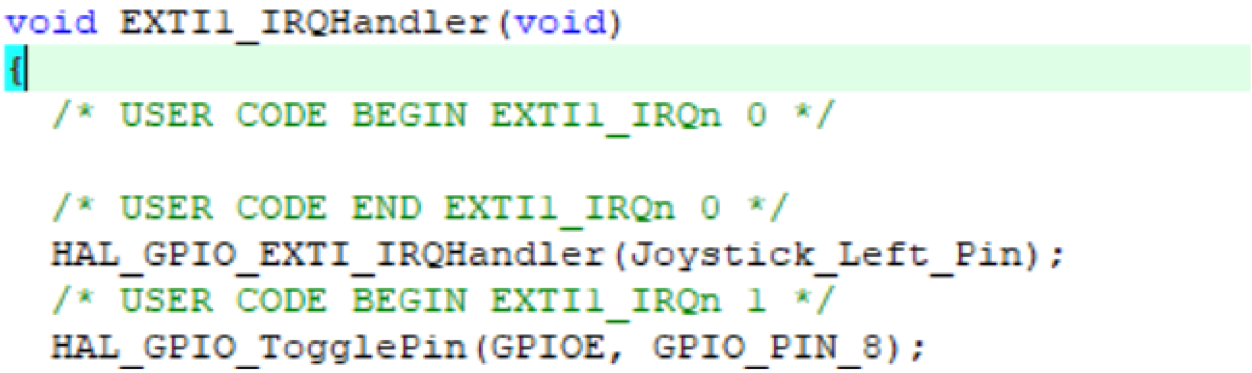
ELEN 121L Lab 4

Joystick Interrupt with LEDs Code:

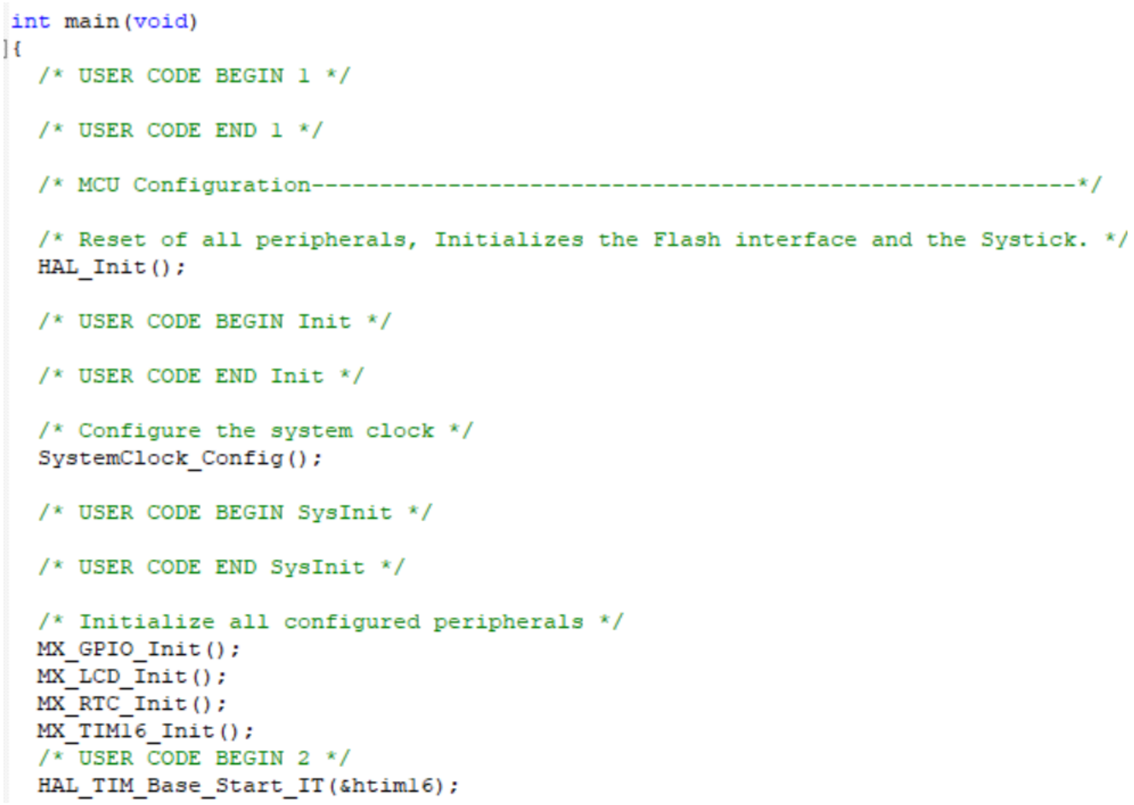
stm32l4xx_it.c code:



```
void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */

  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(Joystick_Center_Pin);
  /* USER CODE BEGIN EXTI0_IRQn 1 */
  stop++;
  HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_2);
  /* USER CODE END EXTI0_IRQn 1 */
}
```



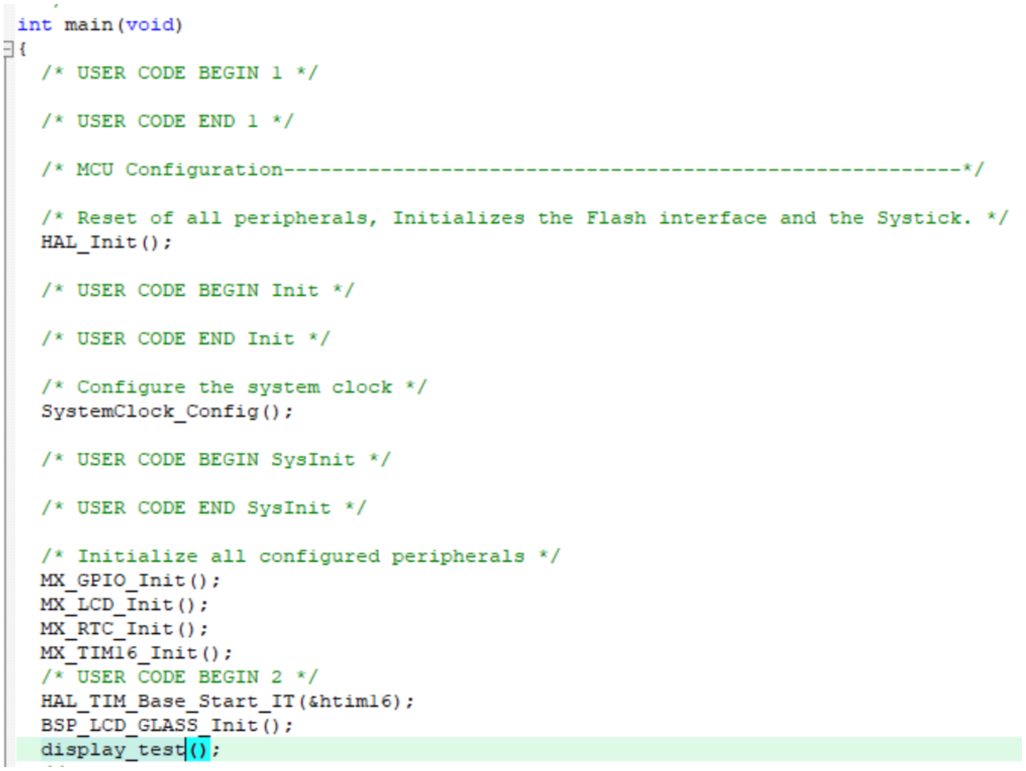
```
void EXTI1_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI1_IRQn 0 */

  /* USER CODE END EXTI1_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(Joystick_Left_Pin);
  /* USER CODE BEGIN EXTI1_IRQn 1 */
  HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_8);
```



```c
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LCD_Init();
  MX_RTC_Init();
  MX_TIM16_Init();
  /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start_IT(&htim16);
```

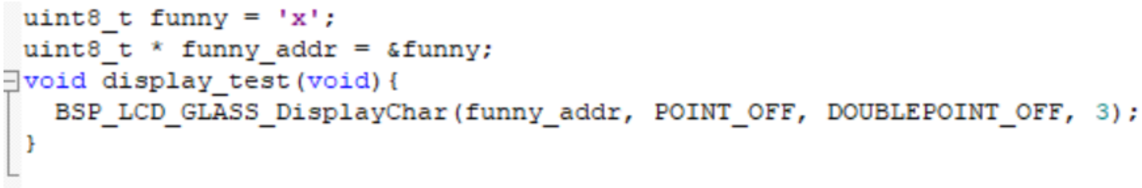
Video for LEDs submitted Separately.

Code for Displaying X:

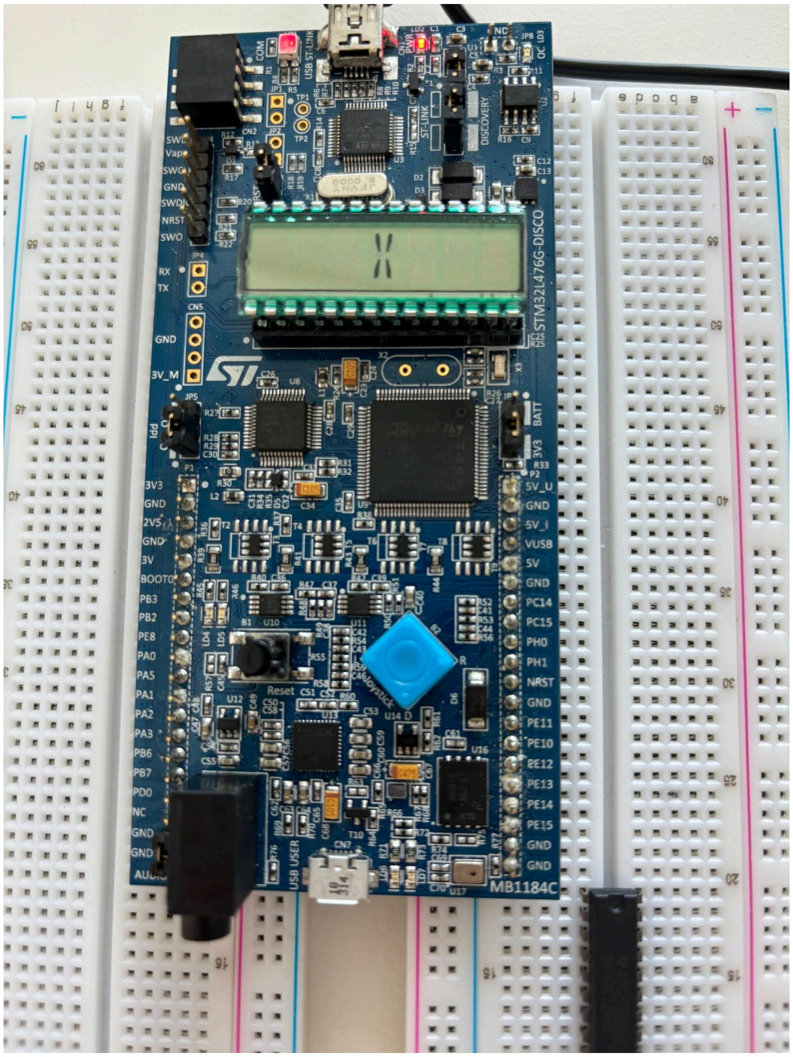


```c
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LCD_Init();
  MX_RTC_Init();
  MX_TIM16_Init();
  /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start_IT(&htim16);
  BSP_LCD_GLASS_Init();
  display_test();
```

display.c code:

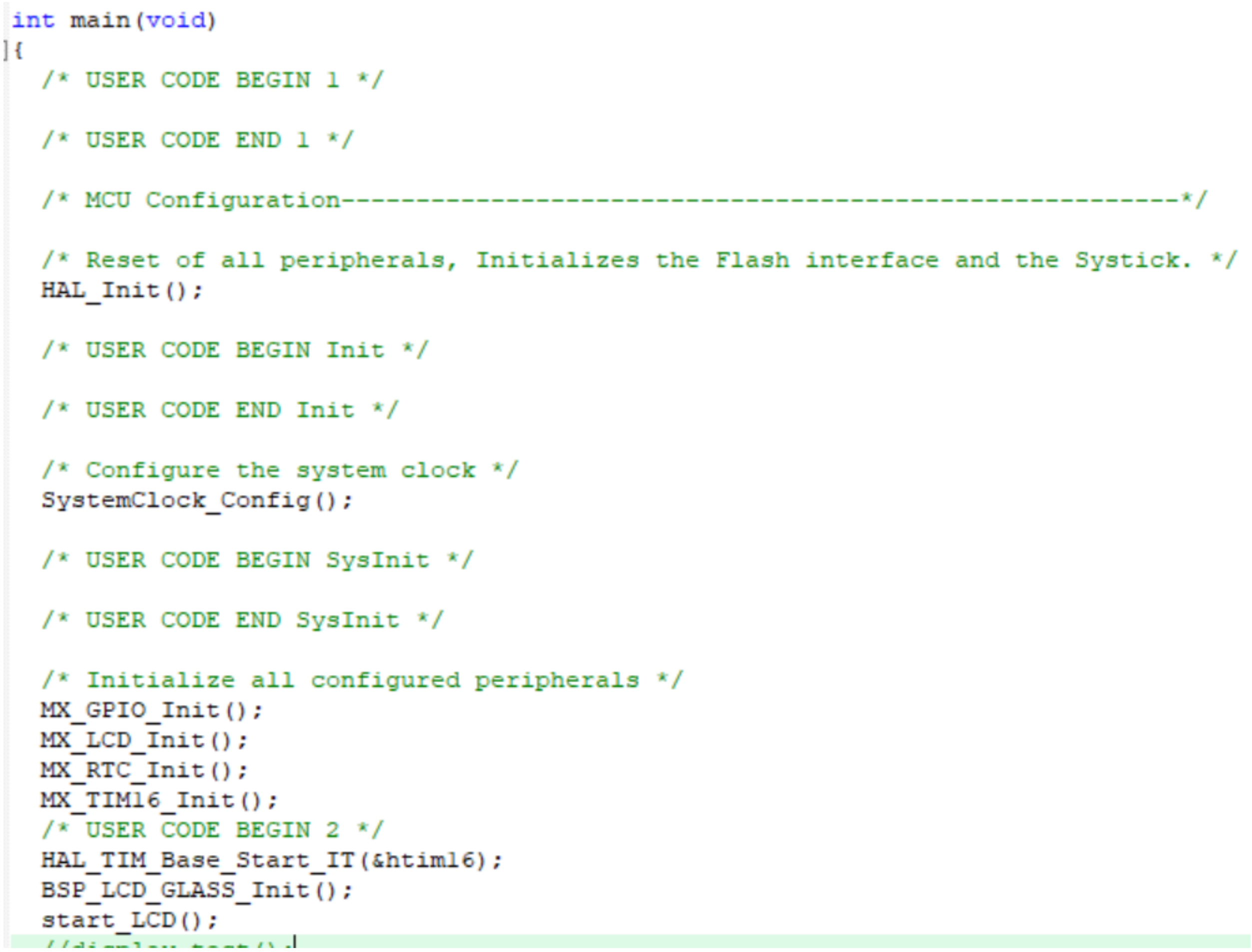


```
uint8_t funny = 'x';
uint8_t * funny_addr = &funny;
void display_test(void){
  BSP_LCD_GLASS_DisplayChar(funny_addr, POINT_OFF, DOUBLEPOINT_OFF, 3);
}
```



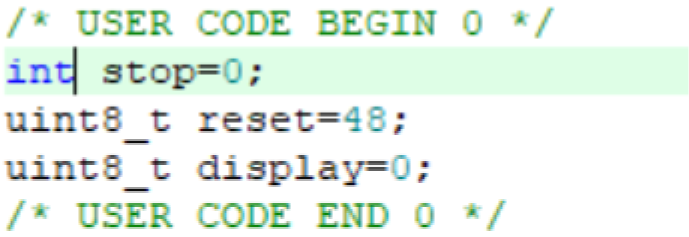
Code for Displaying a Timer:

main.c code:

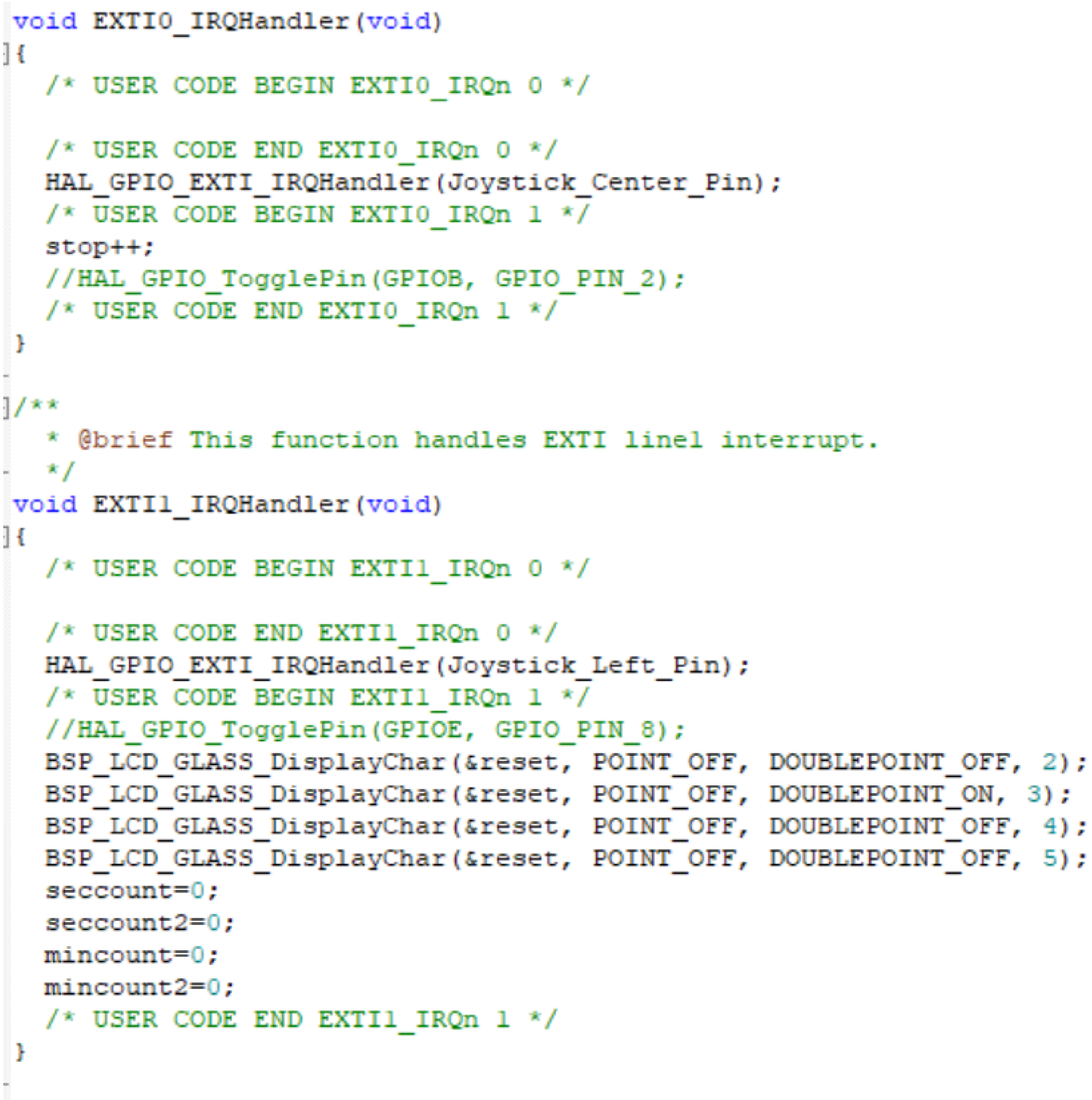


```
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_LCD_Init();
  MX_RTC_Init();
  MX_TIM16_Init();
  /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start_IT(&htim16);
  BSP_LCD_GLASS_Init();
  start_LCD();
  //display_text();
```

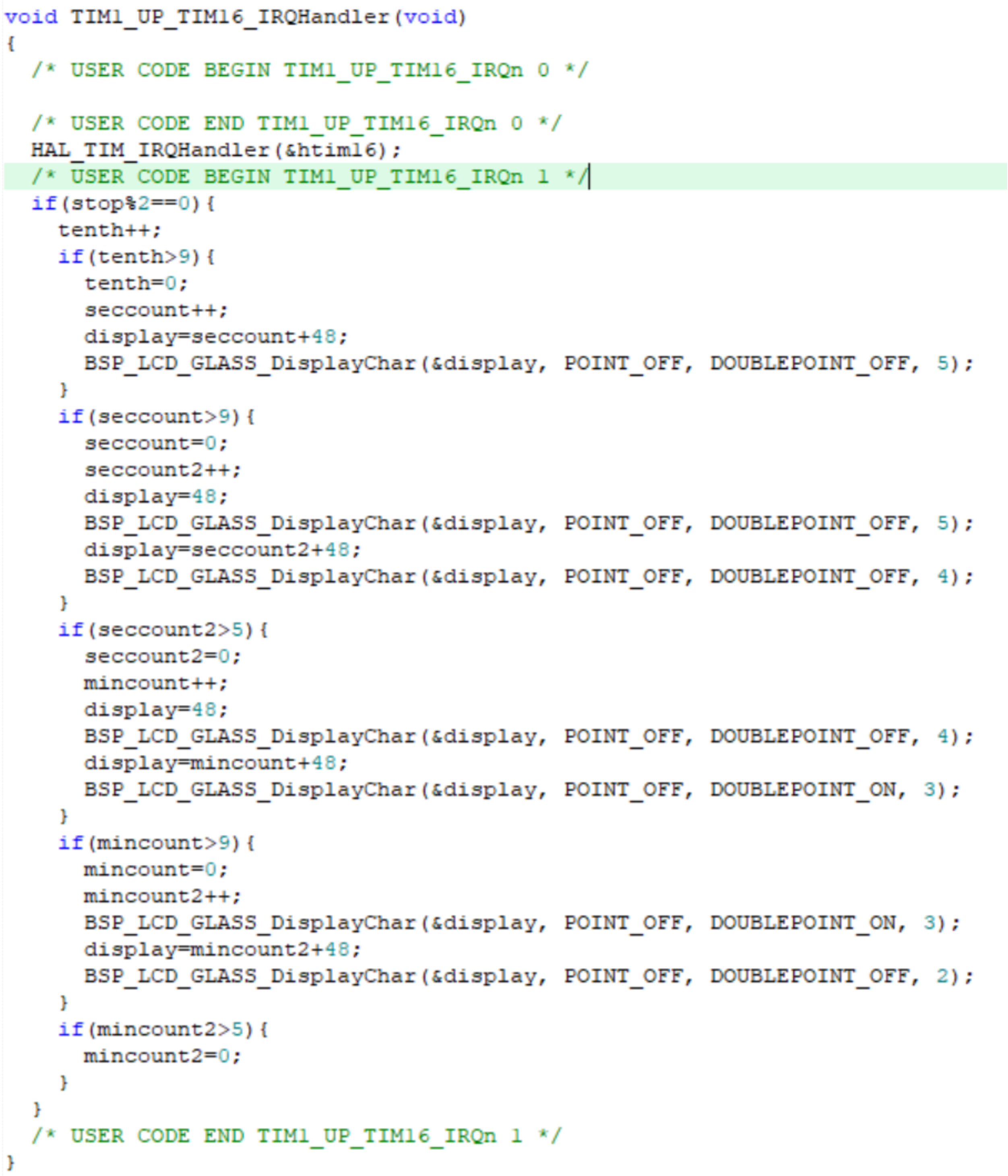
stm32l4xx_it.c code:



```
/* USER CODE BEGIN 0 */
int stop=0;
uint8_t reset=48;
uint8_t display=0;
/* USER CODE END 0 */
```

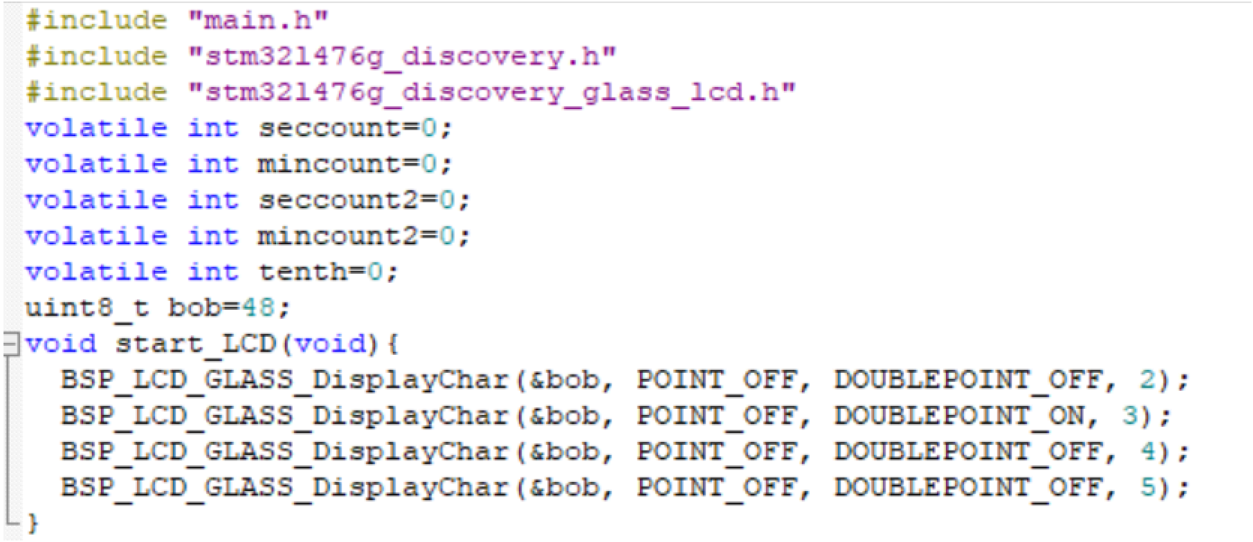


```
void EXTI0_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI0_IRQn 0 */

  /* USER CODE END EXTI0_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(Joystick_Center_Pin);
  /* USER CODE BEGIN EXTI0_IRQn 1 */
  stop++;
  //HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_2);
  /* USER CODE END EXTI0_IRQn 1 */
}

/**
  * @brief This function handles EXTI line1 interrupt.
  */
void EXTI1_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI1_IRQn 0 */

  /* USER CODE END EXTI1_IRQn 0 */
  HAL_GPIO_EXTI_IRQHandler(Joystick_Left_Pin);
  /* USER CODE BEGIN EXTI1_IRQn 1 */
  //HAL_GPIO_TogglePin(GPIOE, GPIO_PIN_8);
  BSP_LCD_GLASS_DisplayChar(&reset, POINT_OFF, DOUBLEPOINT_OFF, 2);
  BSP_LCD_GLASS_DisplayChar(&reset, POINT_OFF, DOUBLEPOINT_ON, 3);
  BSP_LCD_GLASS_DisplayChar(&reset, POINT_OFF, DOUBLEPOINT_OFF, 4);
  BSP_LCD_GLASS_DisplayChar(&reset, POINT_OFF, DOUBLEPOINT_OFF, 5);
  seccount=0;
  seccount2=0;
  mincount=0;
  mincount2=0;
  /* USER CODE END EXTI1_IRQn 1 */
}
```



```
void TIM1_UP_TIM16_IRQHandler(void)
{
  /* USER CODE BEGIN TIM1_UP_TIM16_IRQn 0 */

  /* USER CODE END TIM1_UP_TIM16_IRQn 0 */
  HAL_TIM_IRQHandler(&htim16);
  /* USER CODE BEGIN TIM1_UP_TIM16_IRQn 1 */
  if(stop%2==0){
    tenth++;
    if(tenth>9){
      tenth=0;
      seccount++;
      display=seccount+48;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_OFF, 5);
    }
    if(seccount>9){
      seccount=0;
      seccount2++;
      display=48;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_OFF, 5);
      display=seccount2+48;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_OFF, 4);
    }
    if(seccount2>5){
      seccount2=0;
      mincount++;
      display=48;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_OFF, 4);
      display=mincount+48;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_ON, 3);
    }
    if(mincount>9){
      mincount=0;
      mincount2++;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_ON, 3);
      display=mincount2+48;
      BSP_LCD_GLASS_DisplayChar(&display, POINT_OFF, DOUBLEPOINT_OFF, 2);
    }
    if(mincount2>5){
      mincount2=0;
    }
  }
  /* USER CODE END TIM1_UP_TIM16_IRQn 1 */
}
```

display.c:



```c
#include "main.h"
#include "stm32l476g_discovery.h"
#include "stm32l476g_discovery_glass_lcd.h"
volatile int seccount=0;
volatile int mincount=0;
volatile int seccount2=0;
volatile int mincount2=0;
volatile int tenth=0;
uint8_t bob=48;
void start_LCD(void){
  BSP_LCD_GLASS_DisplayChar(&bob, POINT_OFF, DOUBLEPOINT_OFF, 2);
  BSP_LCD_GLASS_DisplayChar(&bob, POINT_OFF, DOUBLEPOINT_ON, 3);
  BSP_LCD_GLASS_DisplayChar(&bob, POINT_OFF, DOUBLEPOINT_OFF, 4);
  BSP_LCD_GLASS_DisplayChar(&bob, POINT_OFF, DOUBLEPOINT_OFF, 5);
}
```

Timer Video Submitted Separately.