

Learning Report – Embedded C



GLOBAL
ENGINEERING
ACADEMY

Genesis



L&T Technology Services



Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
1	3 rd Jan 2021	Samudrala Apurva		Vivek Kaundal Bhargav N	

CONTENTS

ACTIVITY 1: OBJECT FILE CREATION5

ACTIVITY 2: MINI PROJECT.....5

TABLE OF FIGURES


Figure 1: Linker section.....5

Figure 2: Source Code.....6

Figure 3: Slave code for SPI Protocol.....7

Activity 1: Object File Creation

GitHub Link: <https://github.com/Genesis99003160/Embedded-C>

 MINGW64:/c:/Users/99003160/Documents/Activity 1

```
99003160@EESBLRW120 MINGW64 ~/Documents/Activity 1
$ make
arm-none-eabi-gcc -c -mcpu=cortex-m4 -mthumb -mfloat-abi=soft -std=gnu11 -Wall -
00 -o main1.o main1.c
arm-none-eabi-gcc -c -mcpu=cortex-m4 -mthumb -mfloat-abi=soft -std=gnu11 -Wall -
00 -o cal.o cal.c
arm-none-eabi-gcc -c -mcpu=cortex-m4 -mthumb -mfloat-abi=soft -std=gnu11 -Wall -
00 -o stm32_startup.o stm32_startup.c
arm-none-eabi-gcc -c -mcpu=cortex-m4 -mthumb -mfloat-abi=soft -std=gnu11 -Wall -
00 -o syscalls.o syscalls.c
arm-none-eabi-gcc -mcpu=cortex-m4 -mthumb -mfloat-abi=soft --specs=nano.specs -T
stm32_ls.ld -Wl,-Map=final.map -o final.elf main1.o cal.o stm32_startup.o sysca
lls.o

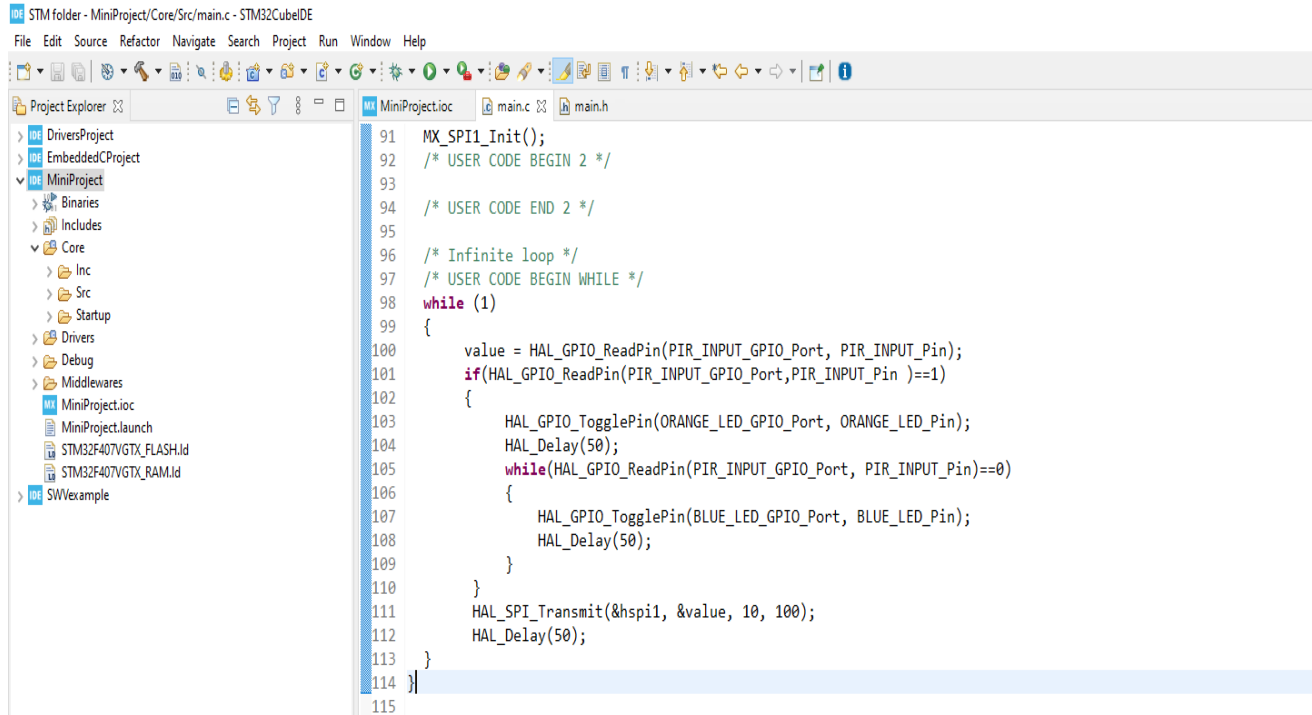
99003160@EESBLRW120 MINGW64 ~/Documents/Activity 1
$ |
```

Figure 1: Object File Creation

Activity 2: Mini Project

Motion Triggered Night Lamp

Link to repository: <https://github.com/Genesis99003160/Embedded-C>



The screenshot shows the STM32CubeIDE interface. The Project Explorer on the left displays the project structure, including the 'MiniProject' folder. The main editor window shows the source code for 'MiniProject.ioc'. The code is as follows:

```
91  MX_SPI1_Init();
92  /* USER CODE BEGIN 2 */
93
94  /* USER CODE END 2 */
95
96  /* Infinite loop */
97  /* USER CODE BEGIN WHILE */
98  while (1)
99  {
100     value = HAL_GPIO_ReadPin(PIR_INPUT_GPIO_Port, PIR_INPUT_Pin);
101     if(HAL_GPIO_ReadPin(PIR_INPUT_GPIO_Port,PIR_INPUT_Pin )==1)
102     {
103         HAL_GPIO_TogglePin(ORANGE_LED_GPIO_Port, ORANGE_LED_Pin);
104         HAL_Delay(50);
105         while(HAL_GPIO_ReadPin(PIR_INPUT_GPIO_Port, PIR_INPUT_Pin)==0)
106         {
107             HAL_GPIO_TogglePin(BLUE_LED_GPIO_Port, BLUE_LED_Pin);
108             HAL_Delay(50);
109         }
110     }
111     HAL_SPI_Transmit(&hspi1, &value, 10, 100);
112     HAL_Delay(50);
113 }
114 }
115 }
```

Figure 2: Source code



```
SPI | Arduino 1.8.13
File Edit Sketch Tools Help

SPI

#include<SPI.h>
volatile boolean val_obtained;
volatile boolean Slave_obtained;

void setup()
{
  Serial.begin(9600);
  pinMode(MISO, OUTPUT);
  pinMode(10, INPUT);
  digitalWrite(10, LOW);
  SPCR |= _BV(SPE);
  val_obtained = false;
  SPI.attachInterrupt();
  Serial.print("SYSTEM STARTING.....");
  delay(2000);
  Serial.print("SYSTEM READY TO START.....");
}
ISR(SPI_STC_vect)
{
  Slave_obtained = SPDR;
  val_obtained = true;
}
void loop()
{
  if (val_obtained)
  {
    if (Slave_obtained == HIGH)
    {
      val_obtained = false;
      Serial.print("\n Night Lamp Triggered\n");
      delay(1000);
    }
  }
  else
  {
    val_obtained = false;
    Serial.print("Night Lamp Switched Off\n");
    delay(1000);
  }
}
```

Figure 3: Slave code for SPI Protocol