Alan Rieger and Ayden Dauenhauer

Prof. Wolfe

ELEN 121L Tuesday 2:15 p.m.

23 April 2024

Lab 3 – Using the LED Strip

Step 2:

GPIO_InitStruct.Pin = SCK_Pin|SDO_Pin;

GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;

GPIO_InitStruct.Pull = GPIO_NOPULL;

GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_MEDIUM;

HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

Line 1 initializes Pin E.13 and Pin E.15.

Line 2 sets the mode to output push pull.

Line 3 sets the pins to no pull.

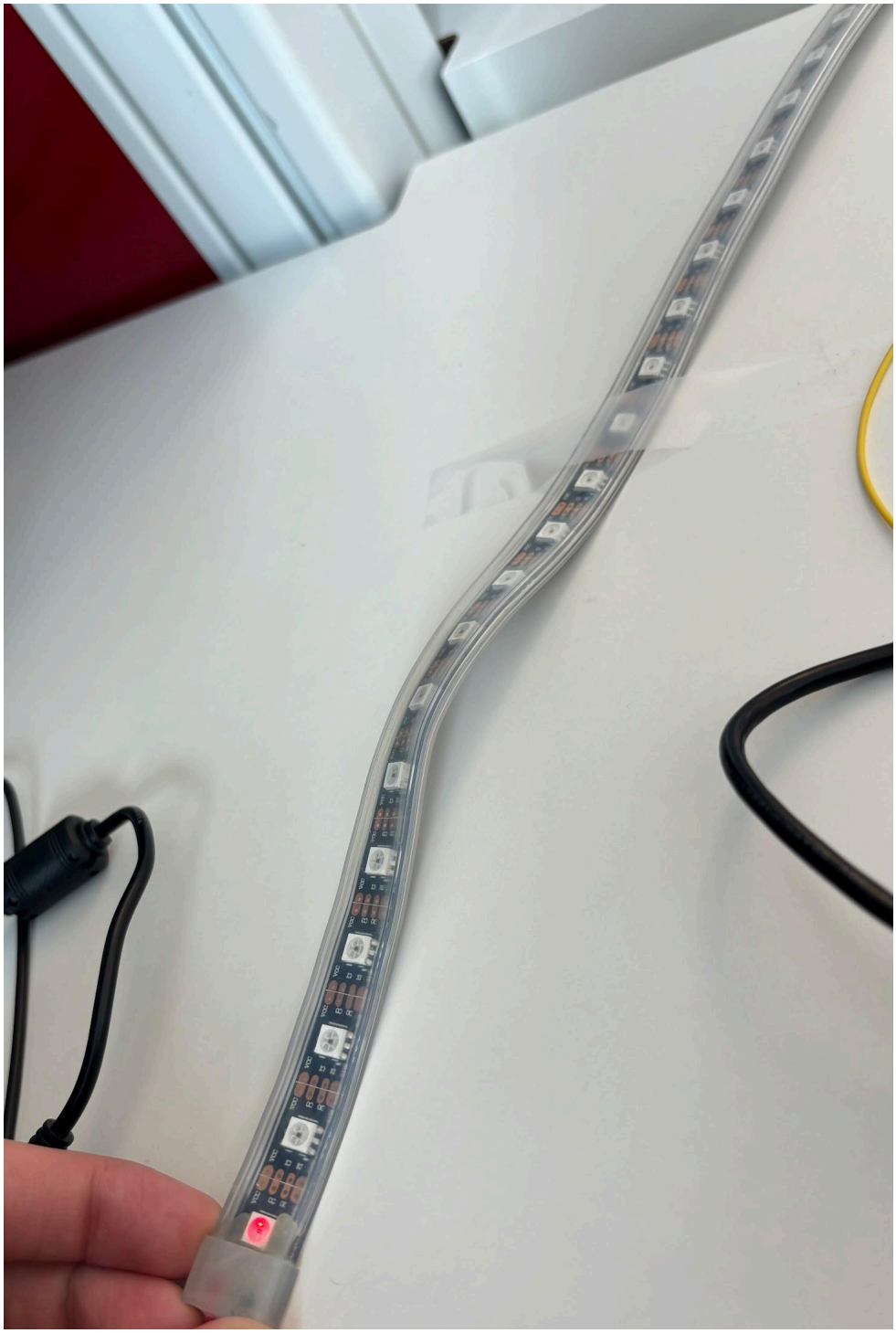
Line 4 sets the speed to medium.

Line 5 gathers the address of the data in pin E.

Step 3:

```
void spi32(unsigned int c){
  int i;
  unsigned int temp;
  unsigned int temp2;
  for(i=0; i<32; i++){
    temp=c;
    temp2=(0x80000000>>i);
    temp&=temp2;
    if(temp!=0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_SET);
    }
    if(temp==0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_RESET);
    }
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_RESET);
  }
}
```

Step 4:

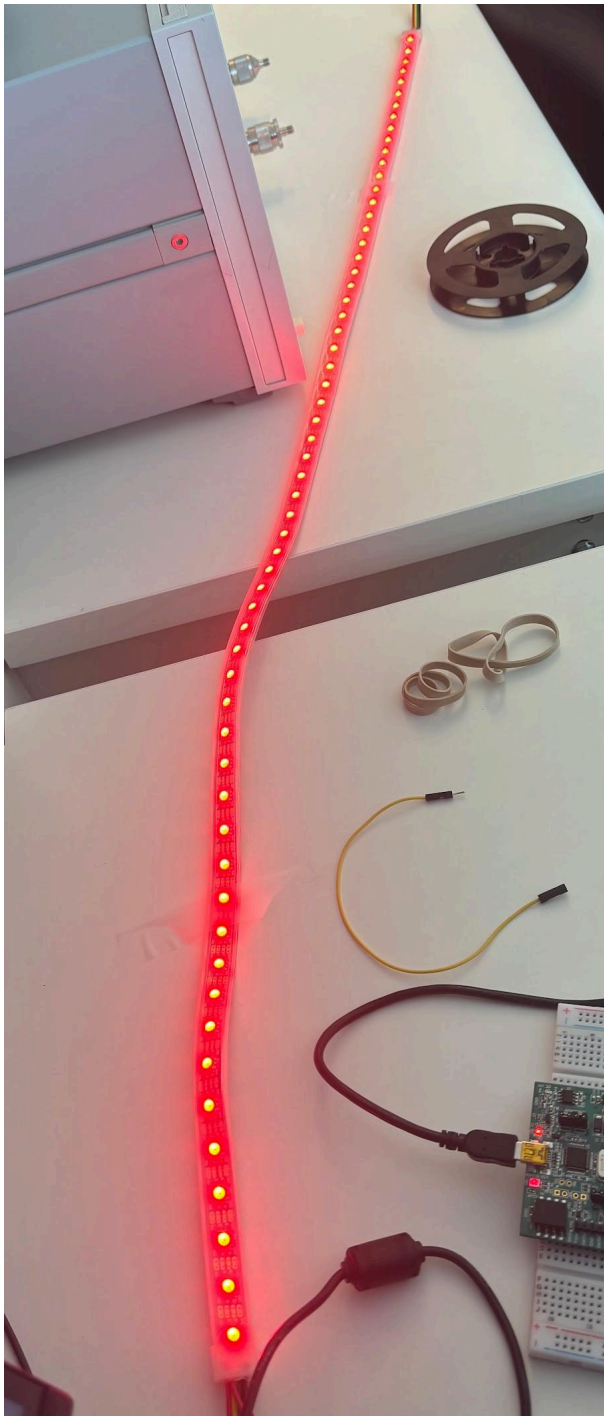


Step 5:

```
void spi32(unsigned int c){
  int i;
  unsigned int temp;
  unsigned int temp2;
  for(i=0; i<32; i++){
    temp=c;
    temp2=(0x80000000>>i);
    temp&=temp2;
    if(temp!=0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_SET);
    }
    if(temp==0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_RESET);
    }
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_RESET);
  }
}
```

```
void sendarray(int buffer[63]){
  int i;
  for(i=0; i<64; i++){
    spi32(buffer[i]);
  }
}
```

```
  /* USER CODE BEGIN 2 */
  int buffer[63];
  int i;
  buffer[0] = 0x00000000;
  buffer[62] = 0xFFFFFFFF;
  for(i=1; i<61; i++){
    buffer[i]=0xF00000FF;
  }
  sendarray(buffer);
  /* USER CODE END 2 */
```



Step 6:

```
void spi32(unsigned int c){
  int i;
  unsigned int temp;
  unsigned int temp2;
  for(i=0; i<32; i++){
    temp=c;
    temp2=(0x80000000>>i);
    temp&=temp2;
    if(temp!=0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_SET);
    }
    if(temp==0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_RESET);
    }
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_RESET);
  }
}
void sendarray(int buffer[63]){
  int i;
  for(i=0; i<64; i++){
    spi32(buffer[i]);
  }
}
```

```
  int buffer[63];
  int i;
  int middle=30;

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    buffer[0] = 0x00000000;
    buffer[62] = 0xFFFFFFFF;
    for(i=1; i<middle; i++){
      buffer[i]=0xF0000000;
    }
    for(i=middle+1; i<62; i++){
      buffer[i]=0xF0000000;
    }
    buffer[middle]=0xF0FF0000;
    sendarray(buffer);
    middle++;
    if(middle==61){
      middle=0;
    }
    HAL_Delay(250);
    /* USER CODE END WHILE */
```

Extra Credit:

```
void spi32(unsigned int c){
  int i;
  unsigned int temp;
  unsigned int temp2;
  for(i=0; i<32; i++){
    temp=c;
    temp2=(0x80000000>>i);
    temp&=temp2;
    if(temp!=0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_SET);
    }
    if(temp==0){
      HAL_GPIO_WritePin(SDO_GPIO_Port, SDO_Pin, GPIO_PIN_RESET);
    }
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(SCK_GPIO_Port, SCK_Pin, GPIO_PIN_RESET);
  }
}
void sendarray(int buffer[63]){
  int i;
  for(i=0; i<64; i++){
    spi32(buffer[i]);
  }
}
```

```
  int buffer[63];
  int i=0;
  int j=1;
  int start=1;
  int colors[7]={0xF00000FF, 0xF000A5FF, 0xF000FFFF, 0xF000FF00, 0xF0FF0000, 0xF082004B, 0xF0FF007F};

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    buffer[0] = 0x00000000;
    buffer[62] = 0xFFFFFFFF;

    for(i=0; i<6; i++){
      for(j=1; j<56; j=j+6) {
        buffer[j]=colors[(i)%6];
        buffer[j+1]=colors[(i+1)%6];
        buffer[j+2]=colors[(i+2)%6];
        buffer[j+3]=colors[(i+3)%6];
        buffer[j+4]=colors[(i+4)%6];
        buffer[j+5]=colors[(i+5)%6];
        buffer[j+6]=colors[(i+6)%6];
      }
      sendarray(buffer);
      HAL_Delay(250);
    }

    start++;
    if(start==61){
      start=1;
    }
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```