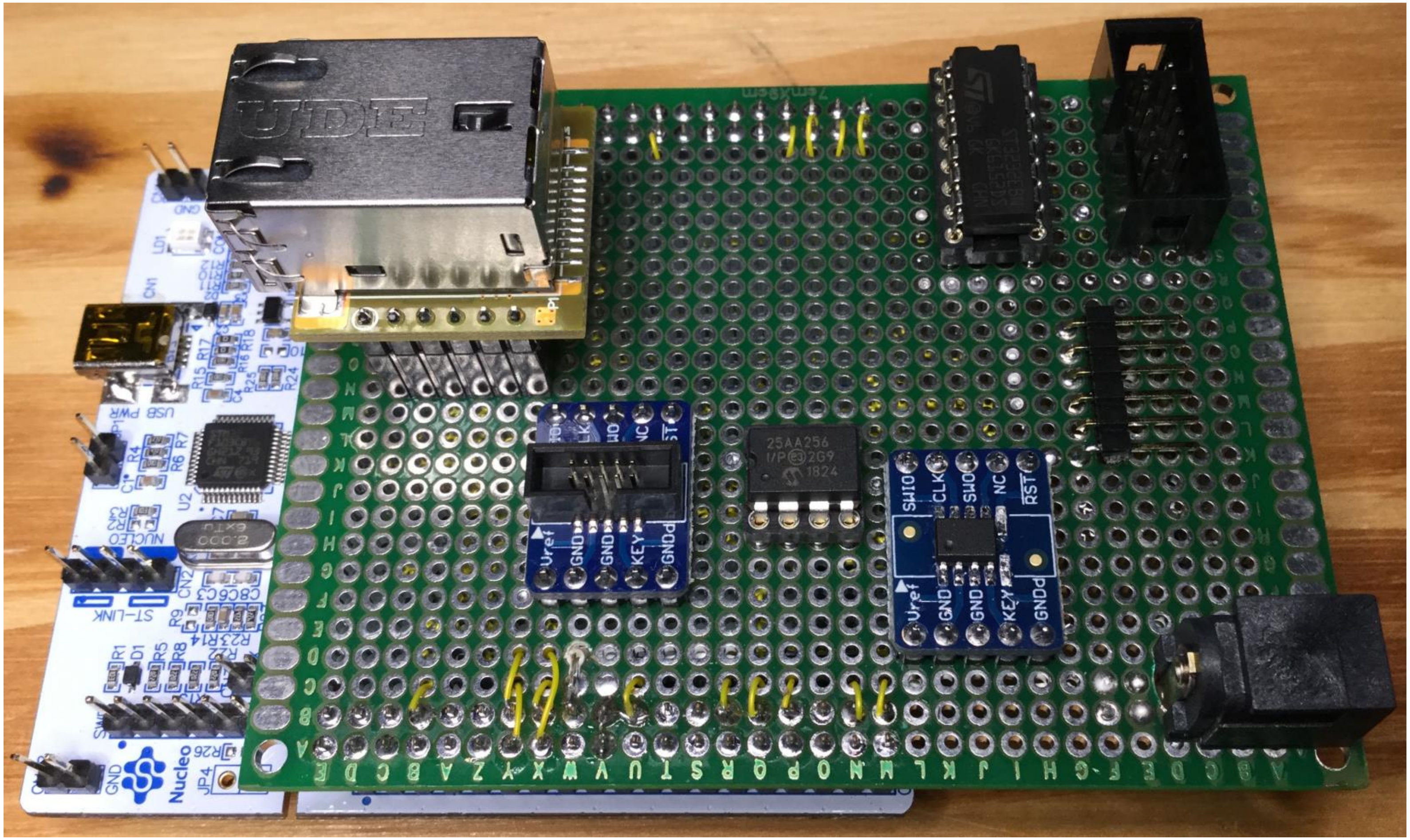
Essential Coding Techniques for Hardware Engineers



Enabling the EEPROM Serial Peripheral Interface

January 30, 2019

Fred Eady

Presented by:

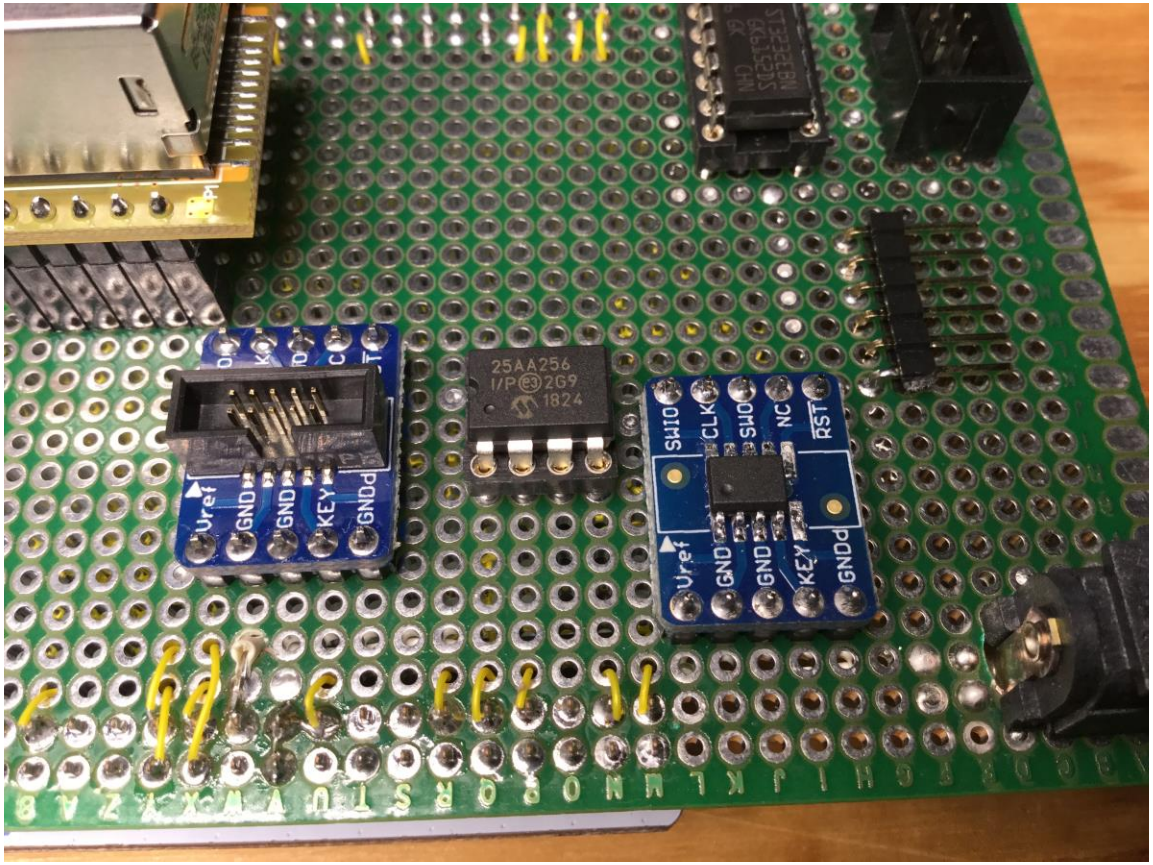
DesignNews

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

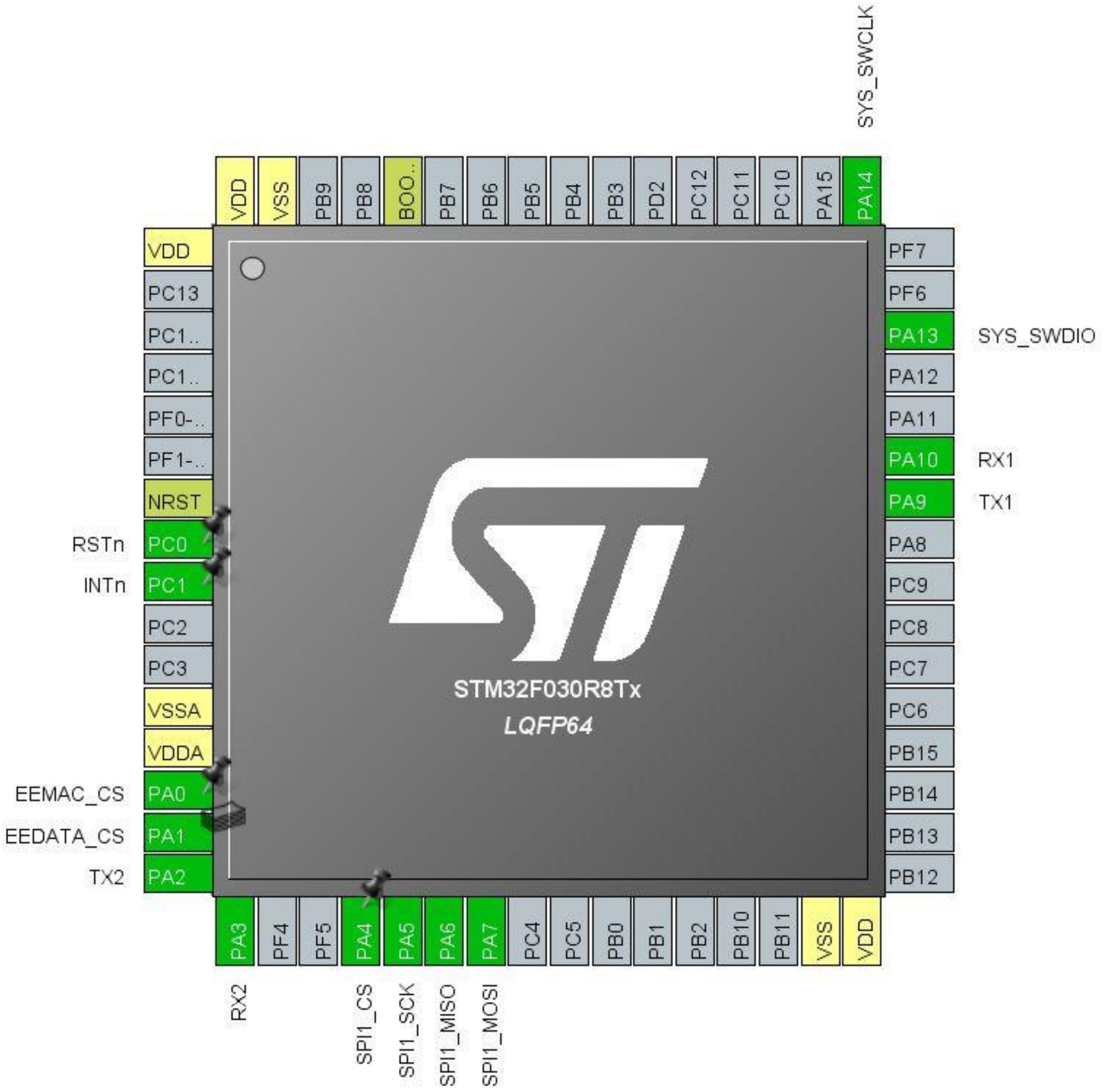
AGENDA

- **Hardware – The SPI EEPROM Interface**
- **Firmware – SPI EEPROM Driver**
- **Day 3 Summary**



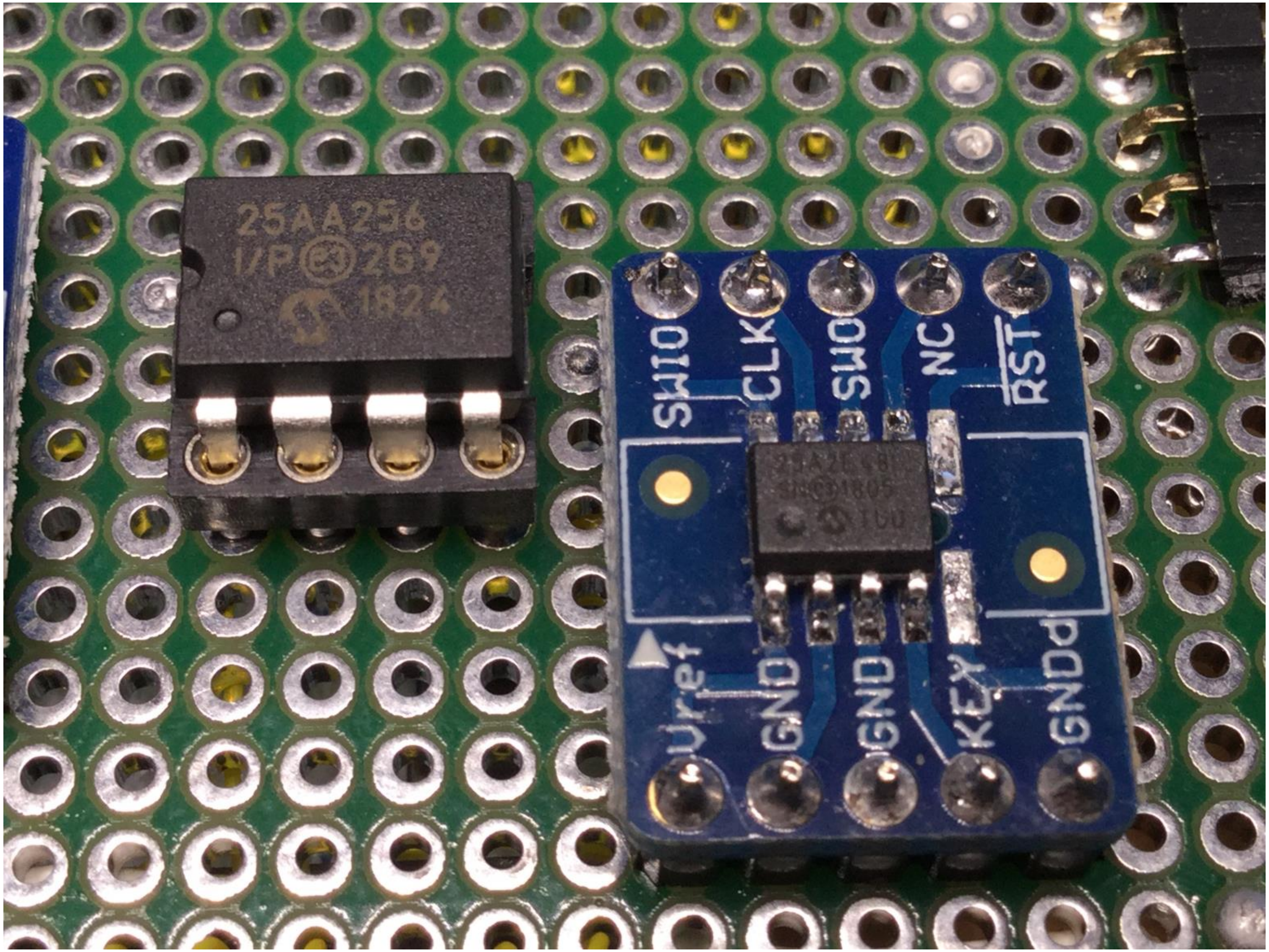
Essential Coding Techniques for Hardware Engineers

Hardware – The SPI EEPROM Interface: ARM



Essential Coding Techniques for Hardware Engineers

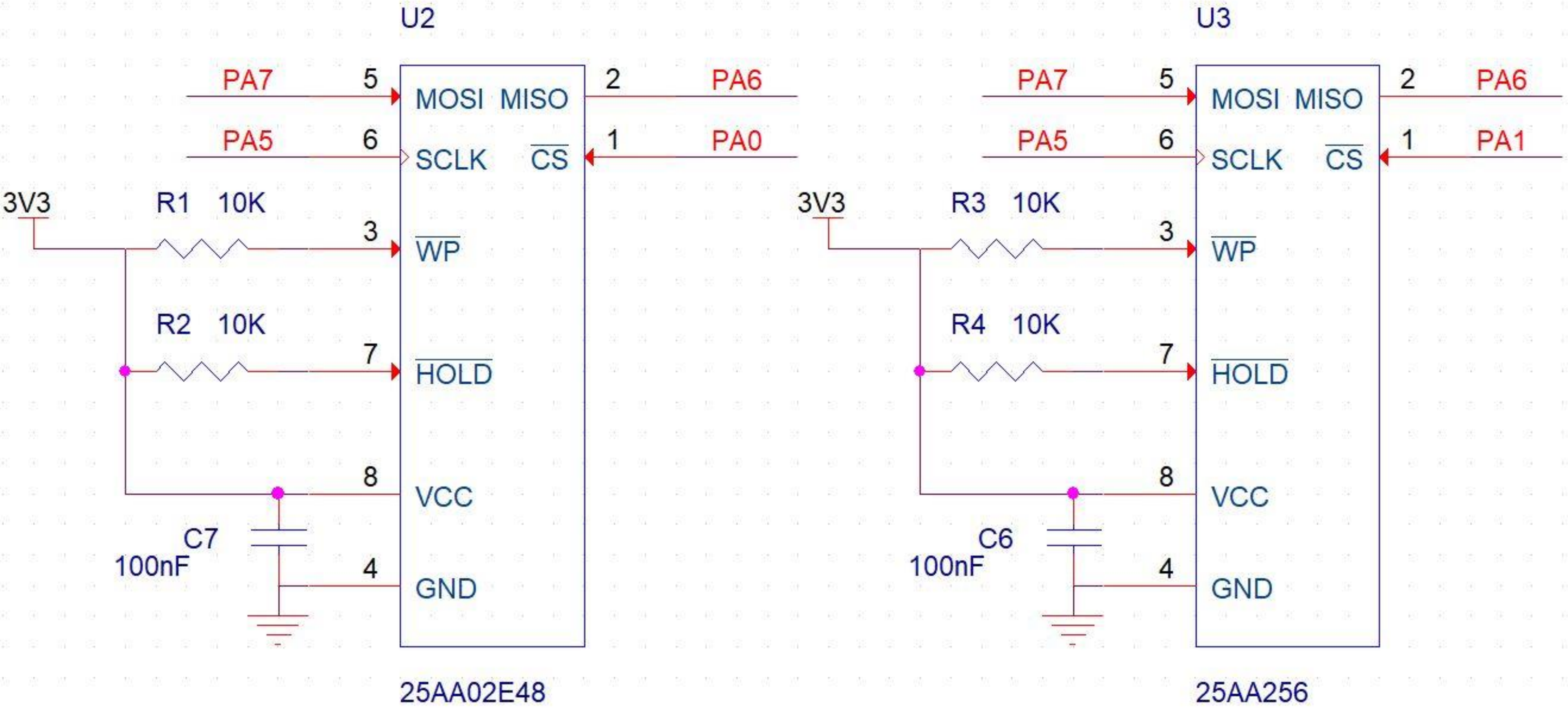
Hardware – The SPI EEPROM Interface: ARM



Presented by:

Essential Coding Techniques for Hardware Engineers

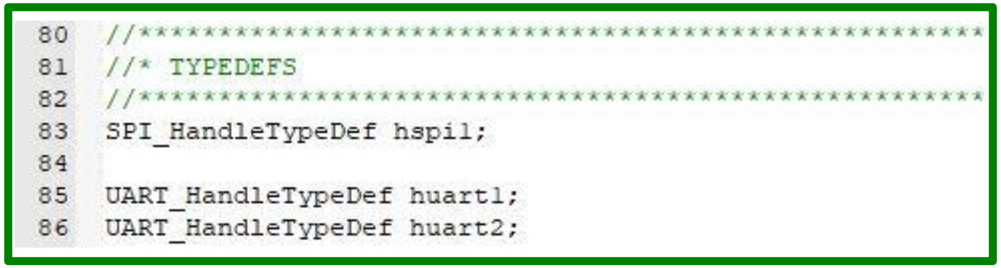
Hardware – The SPI EEPROM Interface: ARM



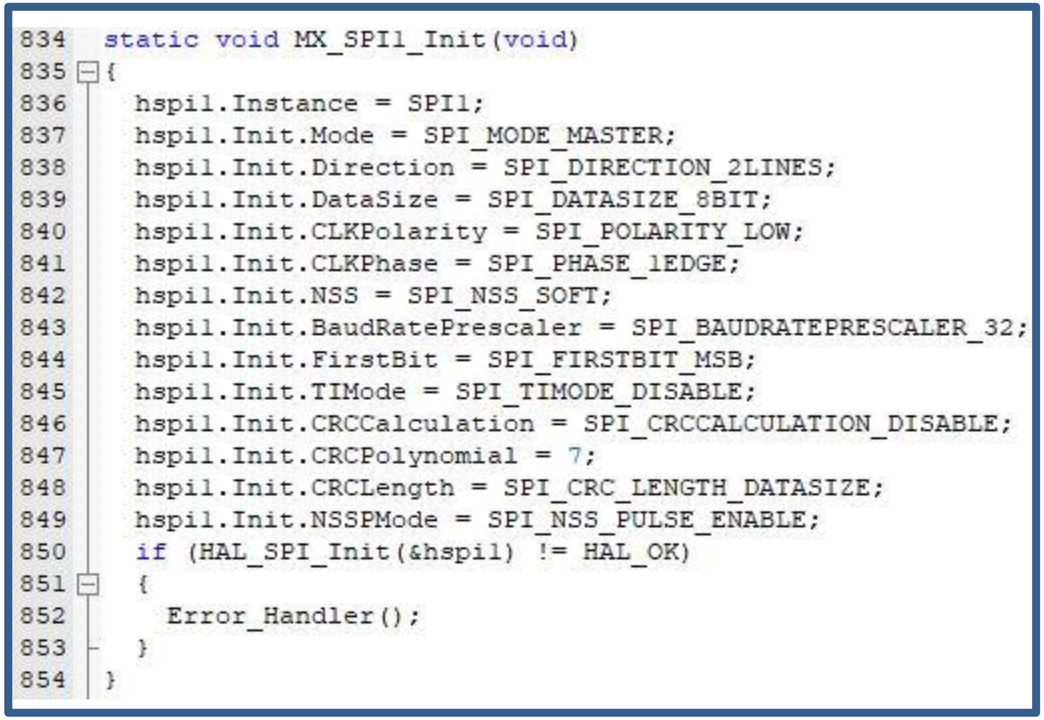
Presented by:

Essential Coding Techniques for Hardware Engineers

Firmware – SPI EEPROM Driver: SPI Init



```c
//****************************************************
//* TYPEDEFS
//****************************************************
SPI_HandleTypeDef hspi1;

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart2;
```



```c
static void MX_SPI1_Init(void)
{
  hspi1.Instance = SPI1;
  hspi1.Init.Mode = SPI_MODE_MASTER;
  hspi1.Init.Direction = SPI_DIRECTION_2LINES;
  hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
  hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
  hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
  hspi1.Init.NSS = SPI_NSS_SOFT;
  hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_32;
  hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi1.Init.CRCPolynomial = 7;
  hspi1.Init.CRCLength = SPI_CRC_LENGTH_DATASIZE;
  hspi1.Init.NSSPMode = SPI_NSS_PULSE_ENABLE;
  if (HAL_SPI_Init(&hspi1) != HAL_OK)
  {
    Error_Handler();
  }
}
```

Presented by:

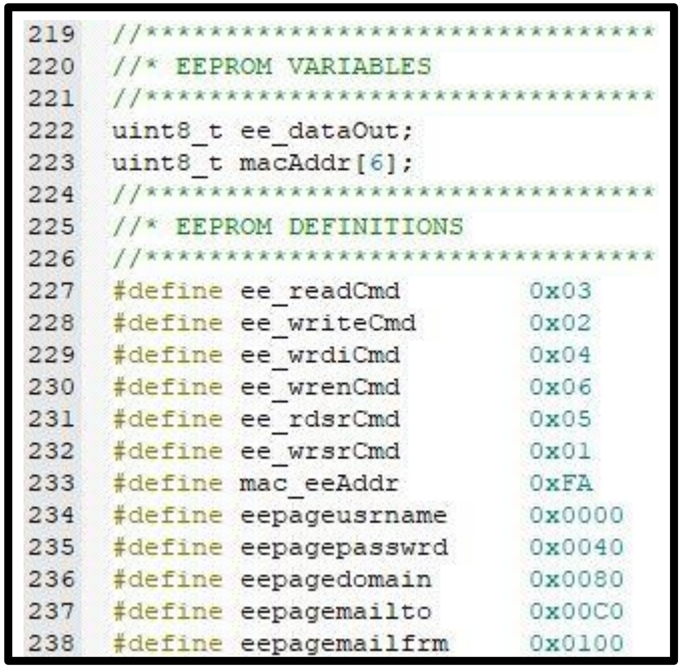
DesignNews

CEC CONTINUING EDUCATION CENTER

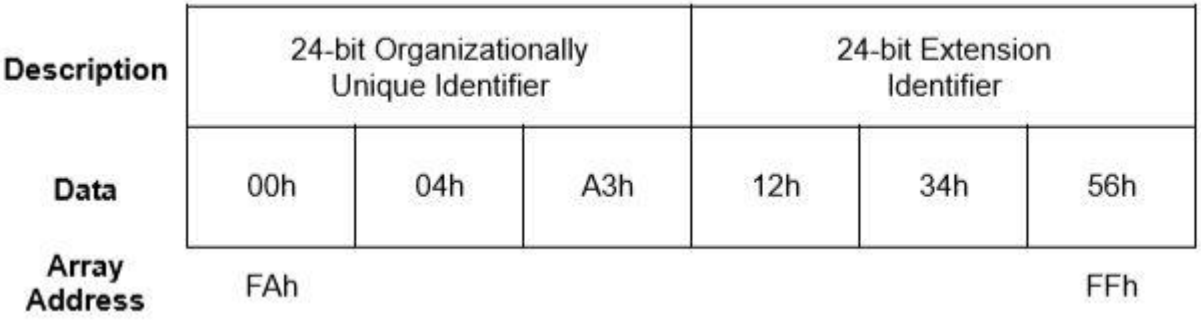
Digi-Key ELECTRONICS

Essential Coding Techniques for Hardware Engineers

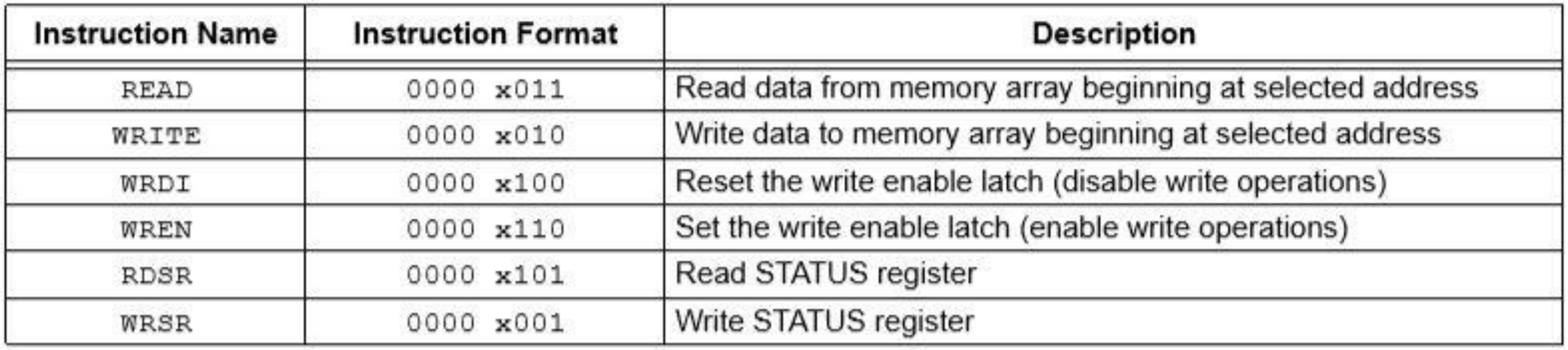
Firmware – SPI EEPROM Driver: EEPROM Variables/Definitions



```
//*******************************
//* EEPROM VARIABLES
//*******************************
uint8_t ee_dataOut;
uint8_t macAddr[6];
//*******************************
//* EEPROM DEFINITIONS
//*******************************
#define ee_readCmd          0x03
#define ee_writeCmd         0x02
#define ee_wrdiCmd          0x04
#define ee_wrenCmd          0x06
#define ee_rdsrCmd          0x05
#define ee_wrsrCmd          0x01
#define mac_eeAddr          0xFA
#define eepageusrname       0x0000
#define eepagepasswrd       0x0040
#define eepagedomain        0x0080
#define eepagemailto        0x00C0
#define eepagemailfrm       0x0100
```



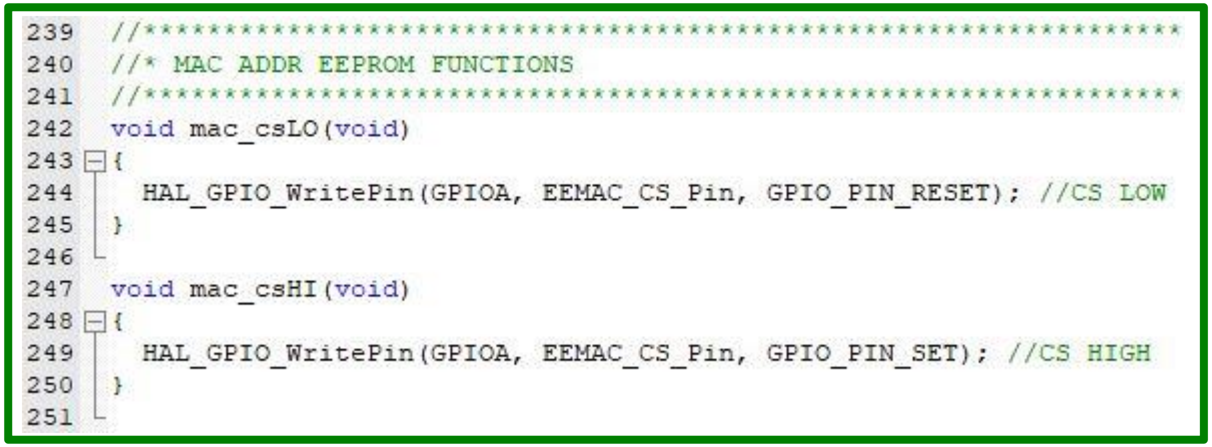
Description	24-bit Organizationally Unique Identifier			24-bit Extension Identifier		
Data	00h	04h	A3h	12h	34h	56h
Array Address	FAh					FFh



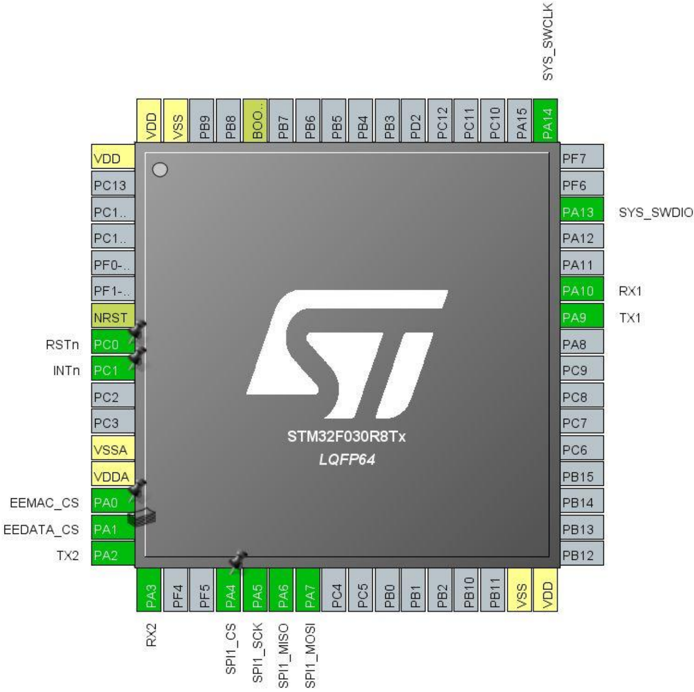
Instruction Name	Instruction Format	Description
READ	0000 x011	Read data from memory array beginning at selected address
WRITE	0000 x010	Write data to memory array beginning at selected address
WRDI	0000 x100	Reset the write enable latch (disable write operations)
WREN	0000 x110	Set the write enable latch (enable write operations)
RDSR	0000 x101	Read STATUS register
WRSR	0000 x001	Write STATUS register

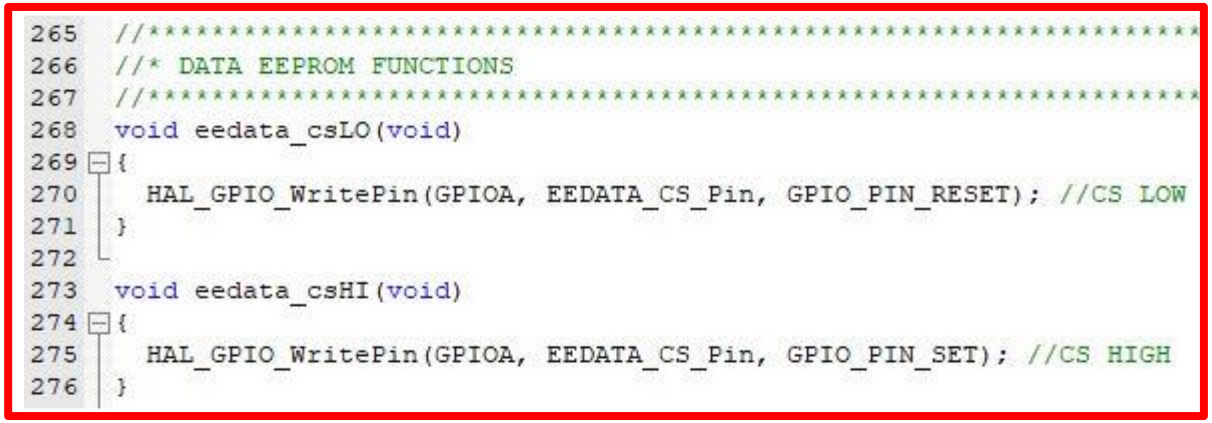
Essential Coding Techniques for Hardware Engineers

Firmware – SPI EEPROM Driver: EEPROM CS Functions



```
//**********************************************************************
//* MAC ADDR EEPROM FUNCTIONS
//**********************************************************************
void mac_csLO(void)
{
  HAL_GPIO_WritePin(GPIOA, EEMAC_CS_Pin, GPIO_PIN_RESET); //CS LOW
}

void mac_csHI(void)
{
  HAL_GPIO_WritePin(GPIOA, EEMAC_CS_Pin, GPIO_PIN_SET); //CS HIGH
}

```

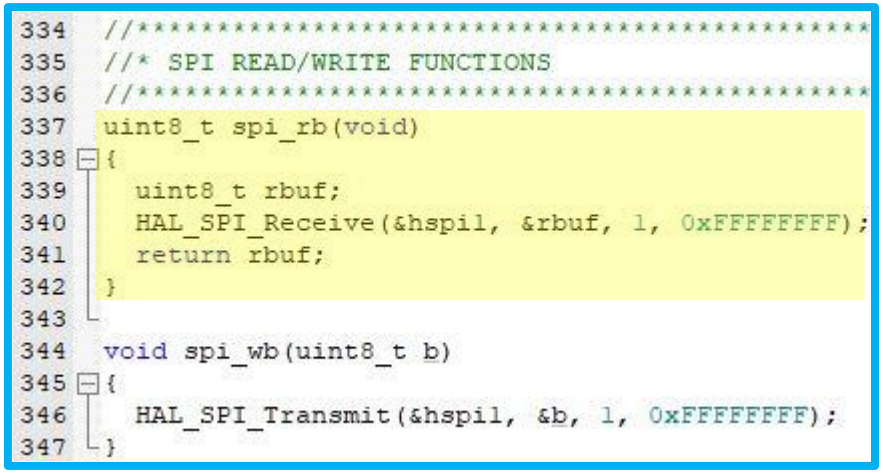




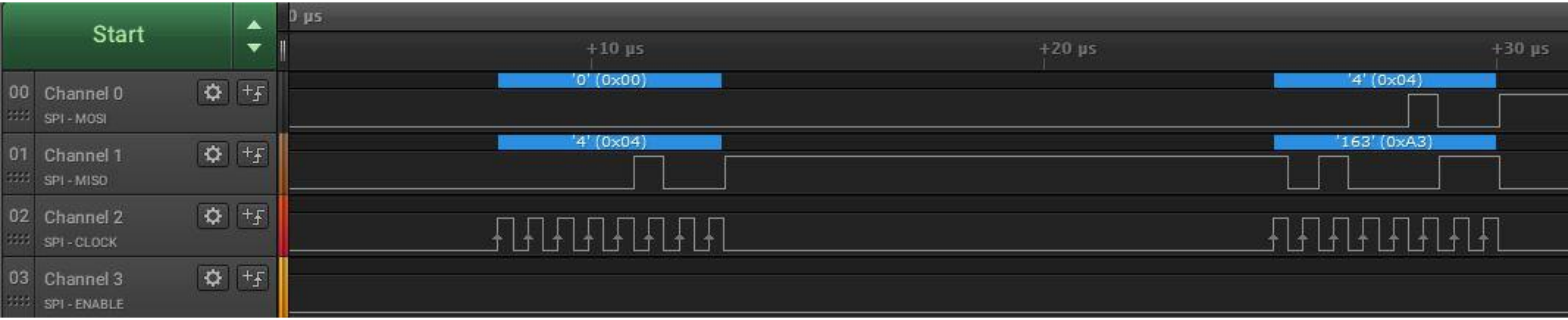
```
//**********************************************************************
//* DATA EEPROM FUNCTIONS
//**********************************************************************
void eedata_csLO(void)
{
  HAL_GPIO_WritePin(GPIOA, EEDATA_CS_Pin, GPIO_PIN_RESET); //CS LOW
}

void eedata_csHI(void)
{
  HAL_GPIO_WritePin(GPIOA, EEDATA_CS_Pin, GPIO_PIN_SET); //CS HIGH
}
```

Essential Coding Techniques for Hardware Engineers

Firmware – SPI EEPROM Driver: SPI Read/Write Functions

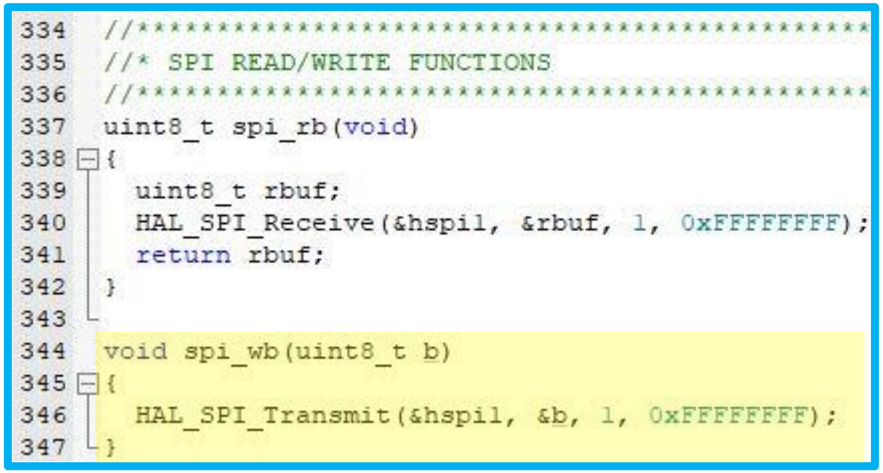


```
//**************************************************
//* SPI READ/WRITE FUNCTIONS
//**************************************************
uint8_t spi_rb(void)
{
   uint8_t rbuf;
   HAL_SPI_Receive(&hspi1, &rbuf, 1, 0xFFFFFFFF);
   return rbuf;
}

void spi_wb(uint8_t b)
{
   HAL_SPI_Transmit(&hspi1, &b, 1, 0xFFFFFFFF);
}
```

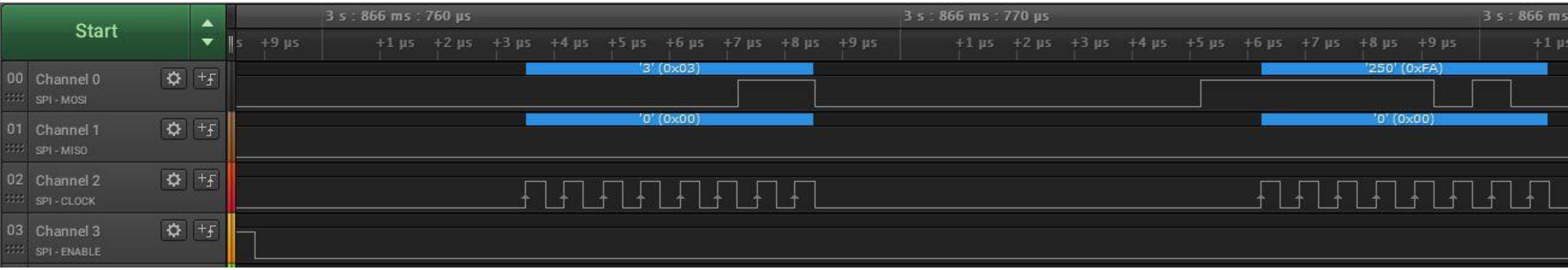


Essential Coding Techniques for Hardware Engineers

Firmware – SPI EEPROM Driver: SPI Read/Write Functions



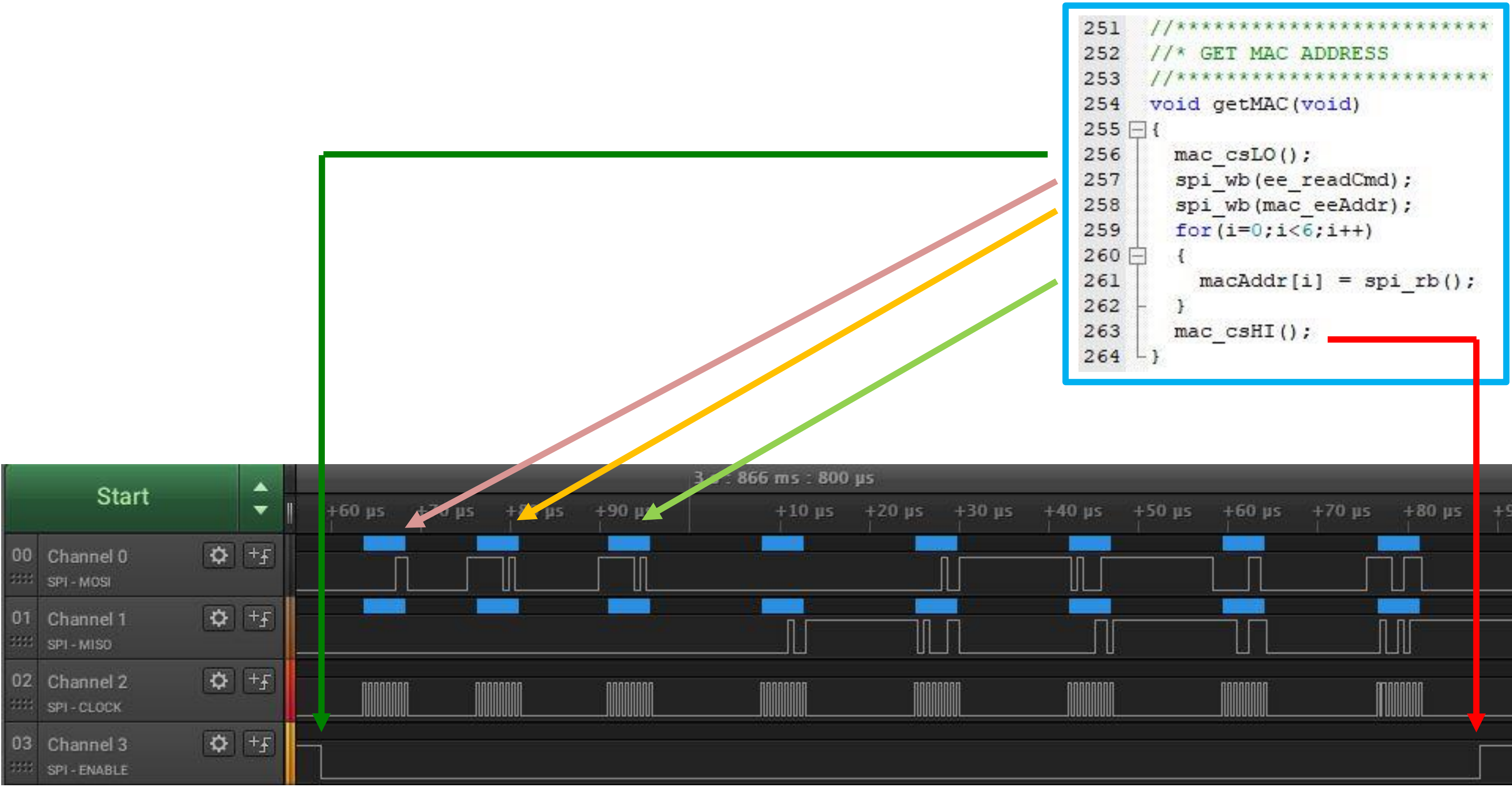
```
//**************************************************
//* SPI READ/WRITE FUNCTIONS
//**************************************************
uint8_t spi_rb(void)
{
   uint8_t rbuf;
   HAL_SPI_Receive(&hspi1, &rbuf, 1, 0xFFFFFFFF);
   return rbuf;
}

void spi_wb(uint8_t b)
{
   HAL_SPI_Transmit(&hspi1, &b, 1, 0xFFFFFFFF);
}
```



Presented by:

Essential Coding Techniques for Hardware Engineers

Firmware – SPI EEPROM Driver: Get MAC Address Function

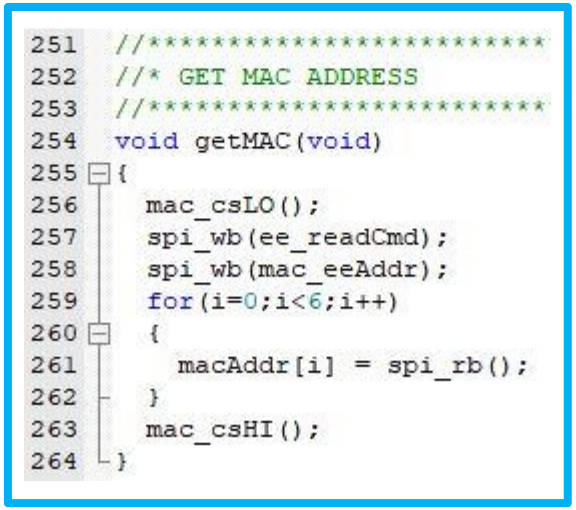


```
//*************************
//* GET MAC ADDRESS
//*************************
void getMAC(void)
{
  mac_csLO();
  spi_wb(ee_readCmd);
  spi_wb(mac_eeAddr);
  for(i=0;i<6;i++)
  {
    macAddr[i] = spi_rb();
  }
  mac_csHI();
}
```

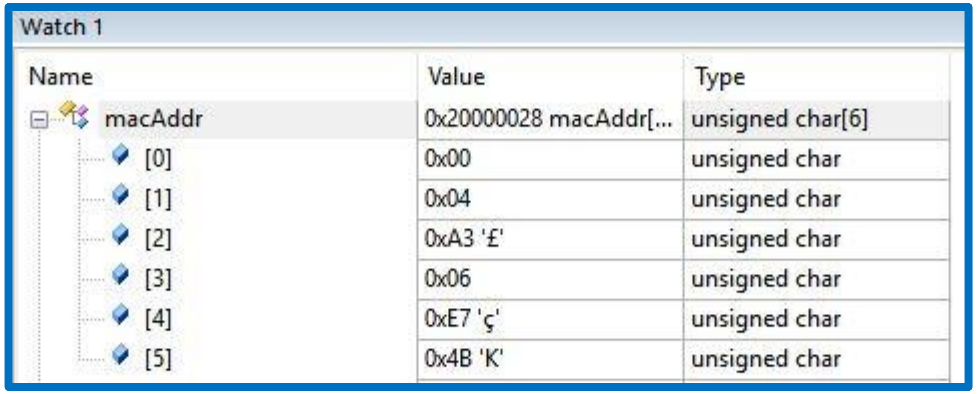
Presented by:

Essential Coding Techniques for Hardware Engineers

Firmware – SPI EEPROM Driver: Get MAC Address Function

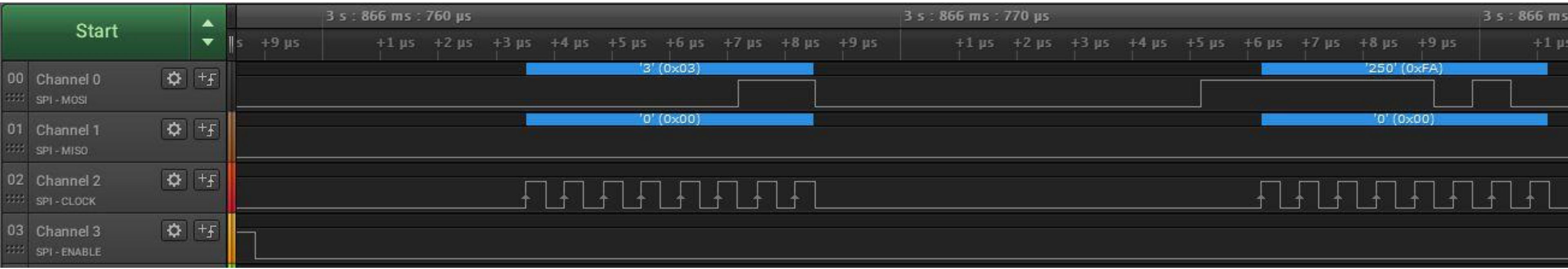


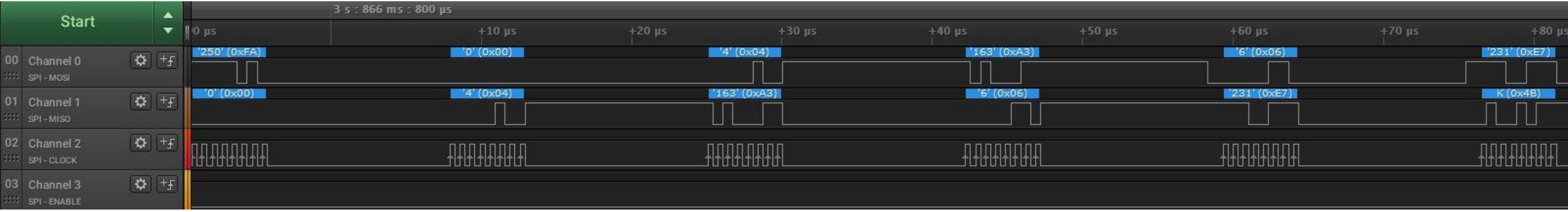
```
//***************************
//* GET MAC ADDRESS
//***************************
void getMAC(void)
{
  mac_csLO();
  spi_wb(ee_readCmd);
  spi_wb(mac_eeAddr);
  for(i=0;i<6;i++)
  {
    macAddr[i] = spi_rb();
  }
  mac_csHI();
}
```



Watch 1

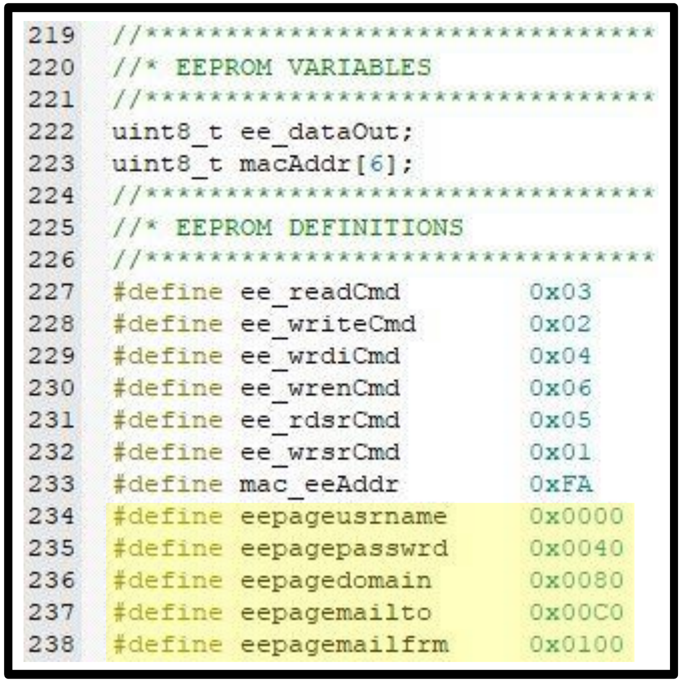
Name	Value	Type
macAddr	0x20000028 macAddr[...	unsigned char[6]
[0]	0x00	unsigned char
[1]	0x04	unsigned char
[2]	0xA3 '£'	unsigned char
[3]	0x06	unsigned char
[4]	0xE7 'ç'	unsigned char
[5]	0x4B 'K'	unsigned char



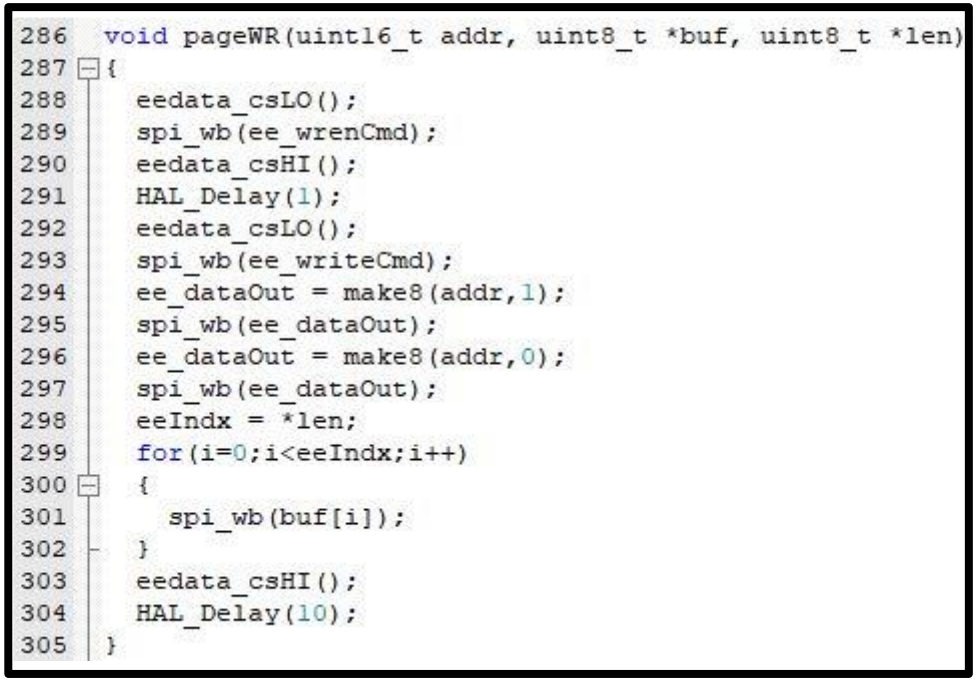


Essential Coding Techniques for Hardware Engineers

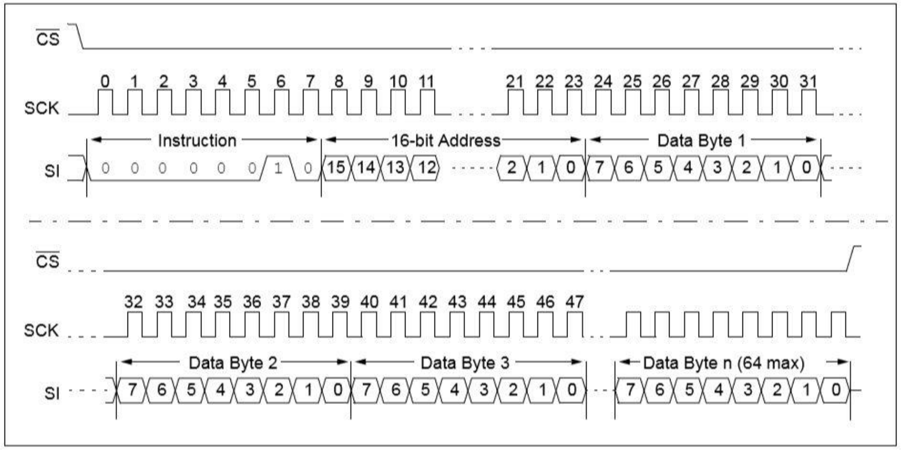
Firmware – SPI EEPROM Driver: Page Write Function



```
//********************************
//* EEPROM VARIABLES
//********************************
uint8_t ee_dataOut;
uint8_t macAddr[6];
//********************************
//* EEPROM DEFINITIONS
//********************************
#define ee_readCmd          0x03
#define ee_writeCmd         0x02
#define ee_wrdiCmd          0x04
#define ee_wrenCmd          0x06
#define ee_rdsrCmd          0x05
#define ee_wrsrCmd          0x01
#define mac_eeAddr          0xFA
#define eepageusrname       0x0000
#define eepagepasswrd       0x0040
#define eepagedomain        0x0080
#define eepagemailto        0x00C0
#define eepagemailfrm       0x0100
```



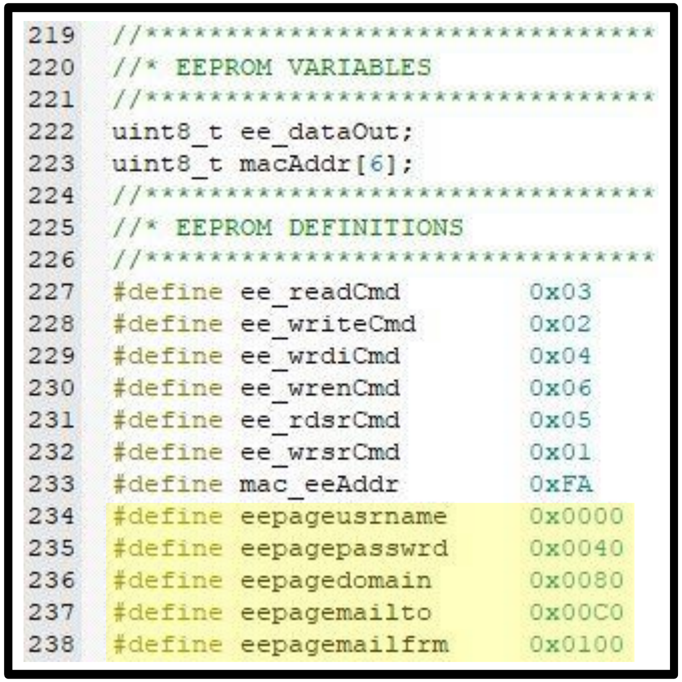
```
void pageWR(uint16_t addr, uint8_t *buf, uint8_t *len)
{
   eedata_csLO();
   spi_wb(ee_wrenCmd);
   eedata_csHI();
   HAL_Delay(1);
   eedata_csLO();
   spi_wb(ee_writeCmd);
   ee_dataOut = make8(addr,1);
   spi_wb(ee_dataOut);
   ee_dataOut = make8(addr,0);
   spi_wb(ee_dataOut);
   eeIndx = *len;
   for(i=0;i<eeIndx;i++)
   {
     spi_wb(buf[i]);
   }
   eedata_csHI();
   HAL_Delay(10);
}
```



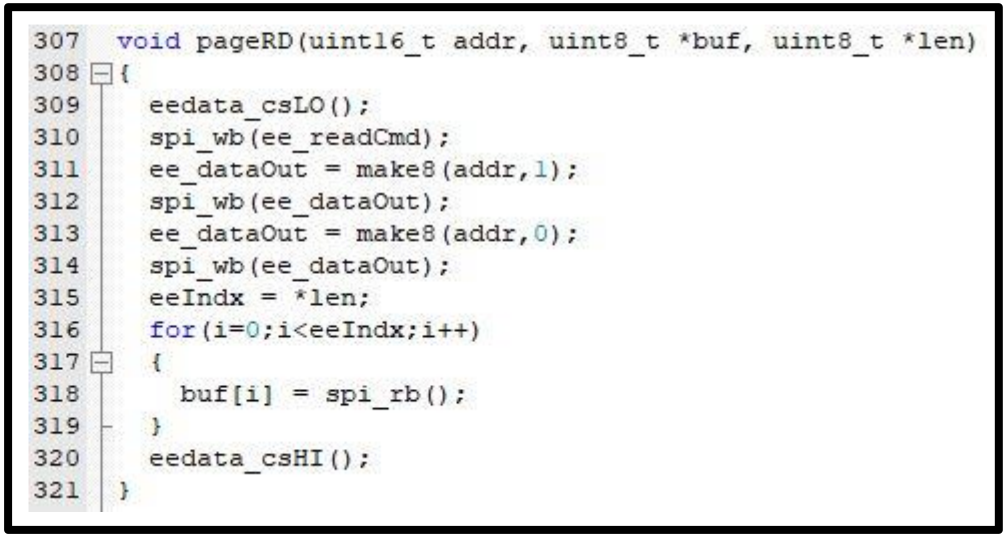
Presented by:

Essential Coding Techniques for Hardware Engineers

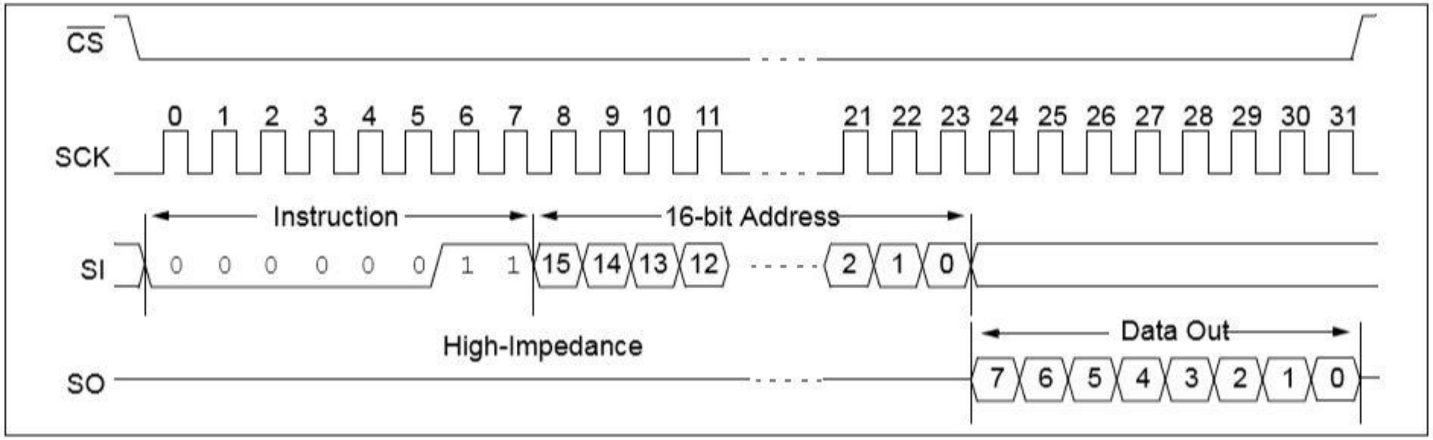
Firmware – SPI EEPROM Driver: Page Read Function



```
//*******************************
//* EEPROM VARIABLES
//*******************************
uint8_t ee_dataOut;
uint8_t macAddr[6];
//*******************************
//* EEPROM DEFINITIONS
//*******************************
#define ee_readCmd            0x03
#define ee_writeCmd           0x02
#define ee_wrdiCmd            0x04
#define ee_wrenCmd            0x06
#define ee_rdsrCmd            0x05
#define ee_wrsrCmd            0x01
#define mac_eeAddr            0xFA
#define eepageusrname         0x0000
#define eepagepasswrd         0x0040
#define eepagedomain          0x0080
#define eepagemailto          0x00C0
#define eepagemailfrm         0x0100
```



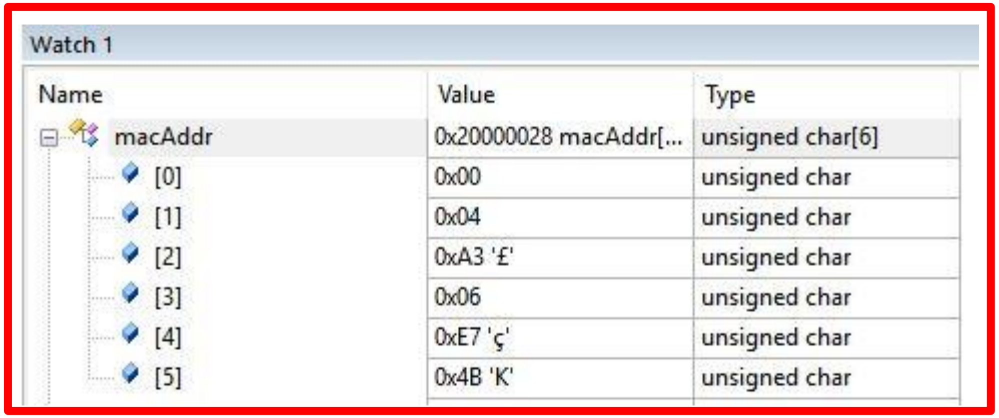
```
void pageRD(uint16_t addr, uint8_t *buf, uint8_t *len)
{
   eedata_csLO();
   spi_wb(ee_readCmd);
   ee_dataOut = make8(addr,1);
   spi_wb(ee_dataOut);
   ee_dataOut = make8(addr,0);
   spi_wb(ee_dataOut);
   eeIndx = *len;
   for(i=0;i<eeIndx;i++)
   {
     buf[i] = spi_rb();
   }
   eedata_csHI();
}
```



Presented by:

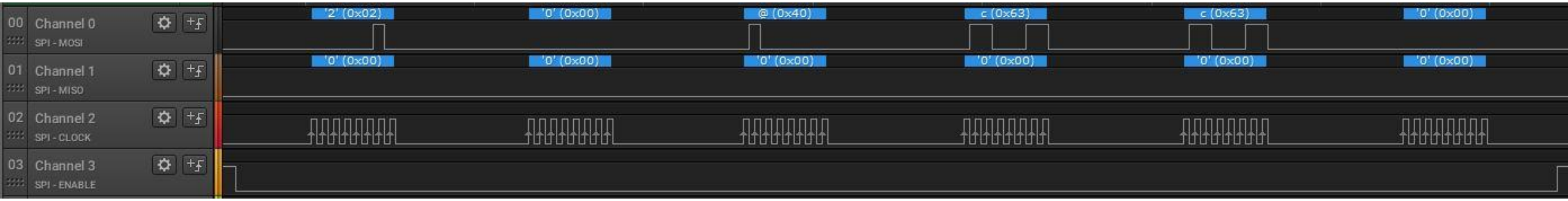
Essential Coding Techniques for Hardware Engineers

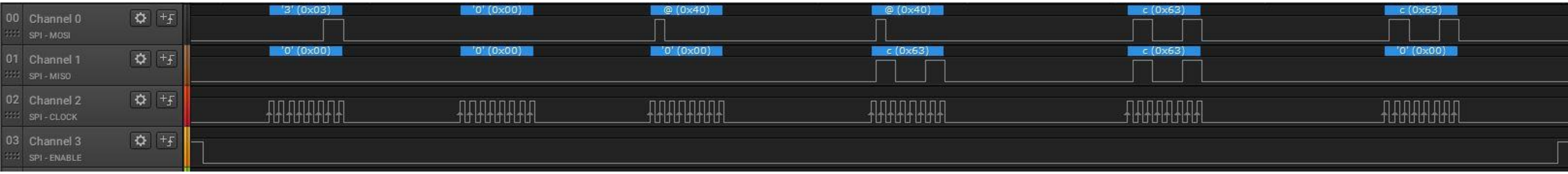
Day 3 Summary



Watch 1

Name	Value	Type
macAddr	0x20000028 macAddr[...	unsigned char[6]
[0]	0x00	unsigned char
[1]	0x04	unsigned char
[2]	0xA3 '£'	unsigned char
[3]	0x06	unsigned char
[4]	0xE7 'ç'	unsigned char
[5]	0x4B 'K'	unsigned char





Presented by:

Essential Coding Techniques for Hardware Engineers

A Peek At What's To Come