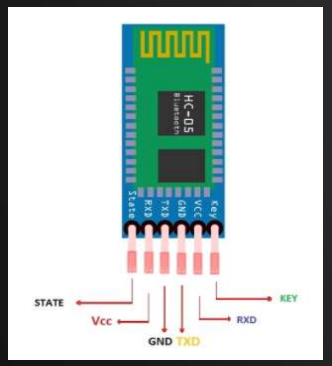

LIVE
LEARN
EMBEDDED SYSTEM
ARM CORTEX M4
DAY 28/30
(30 DAYS CHALLENGE)
M K JEEVARAJAN

What you will Learn Today?

- ✓ Bluetooth Based Home Automation using STM32F411

Bluetooth Control Home Automation

Hardware Required



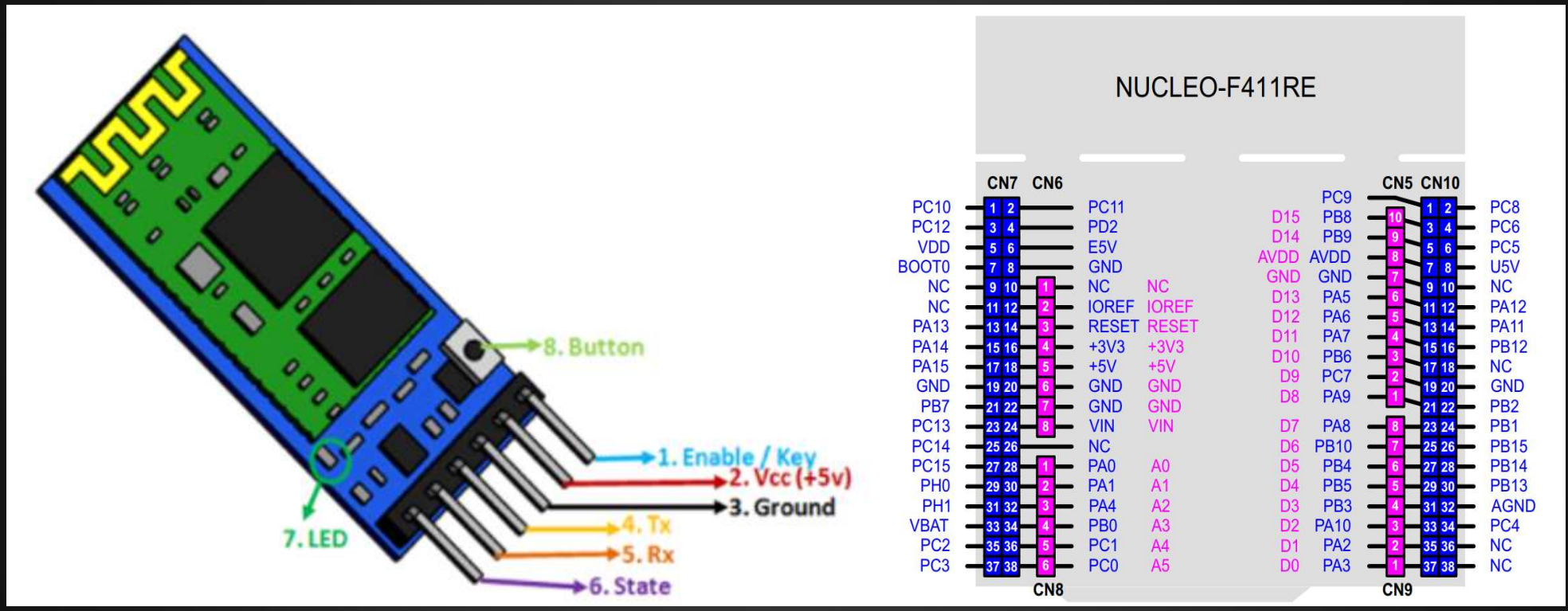
SOFTWARE REQUIRED

- STM2CUBE IDE
- SALEA LOGIC ANALYZER
- Android Bluetooth control(PLAY STORE)

Advantages of Bluetooth

- Advantages
  - ✓Bluetooth: interoperable
  - ✓IrDA: line of sight needed, point-to-point
  - ✓WLAN: higher power consumption
- Disadvantages
  - ✓Bluetooth: only up to 1 Mbps
  - ✓IrDA: much cheaper, faster (up to 16 Mbps)
  - ✓WLAN: faster (up to 11 Mbps)

SCHEMATIC DESIGN



//PB6->TX

//PB7->RX ->BLUETOOTH TX

//RELAY -PA5

HC-05 Default Settings

Default Bluetooth Name: “HC-05”

Default Password: 1234 or 0000

Default Communication: Slave

Default Mode: Data Mode

Data Mode Baud Rate: 9600, 8, N, 1

Command Mode Baud Rate: 38400, 8, N, 1

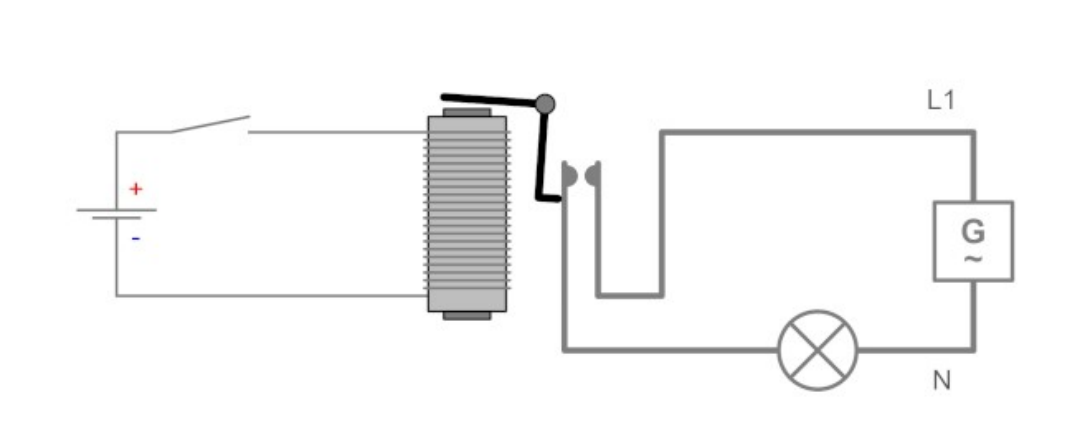
Default firmware: LINVOR

Tech Specifications

- Serial Bluetooth module for arduino other microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

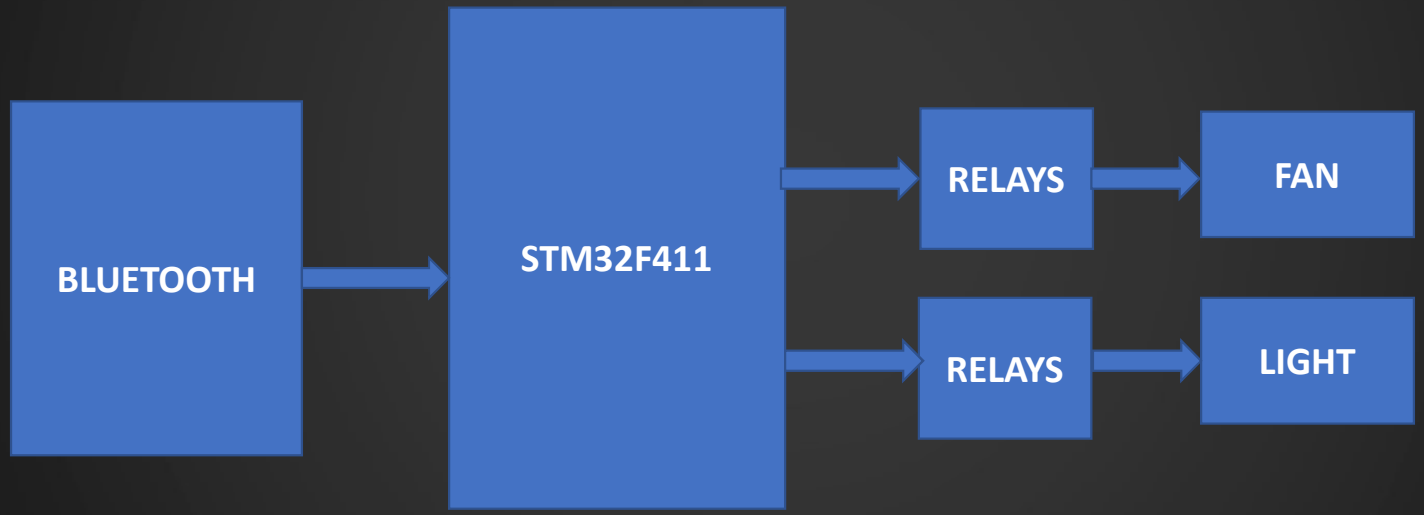
What is a Relay

A relay is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals.



The traditional form of a relay uses an electromagnet to close or open the contacts, but other operating principles have been invented, such as in solid-state relays which use semiconductor properties for control without relying on moving parts.

Block Diagram



Serial Data

	DATA	OPERATION
Fan on	A	Fan Turned ON
Fan off	B	Fan Turned OFF

UART Parameters

- Baud Rate (9600, 19200, 115200, others)
- Number of Data Bits (7, 8)
- Parity Bit (On, Off)
- Stop Bits (0, 1, 2)
- Flow Control (None, On, Hardware)

Connection Details

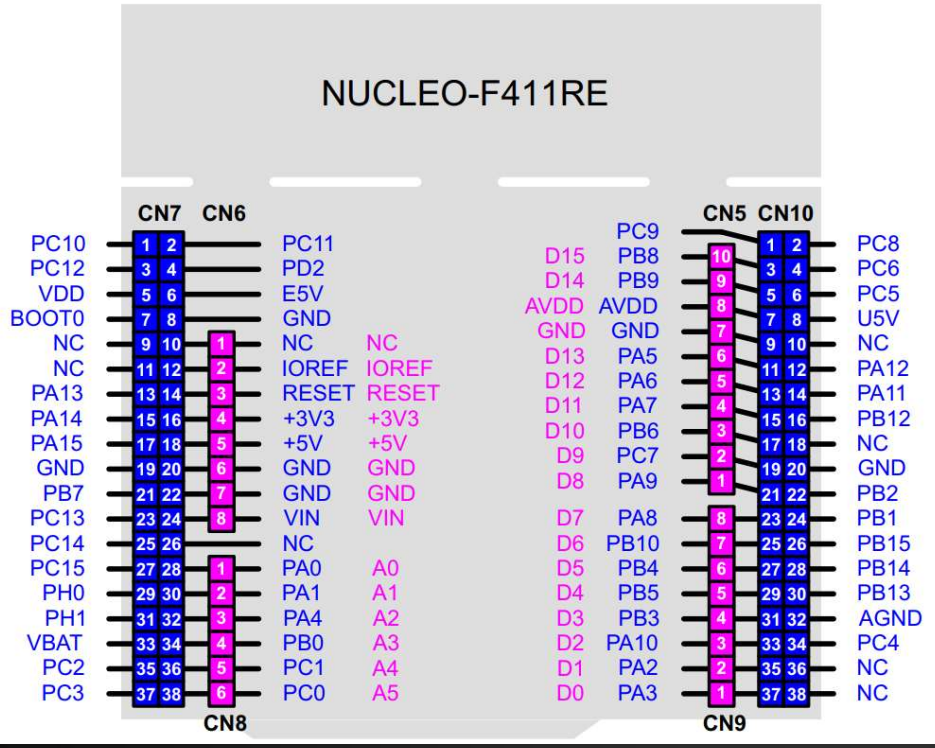


Table 9. Alternate function mapping (continued)

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/I2S1S PI2/ I2S2/SPI3/ I2S3	SPI2/I2S2/ SPI3/ I2S3/SPI4/ I2S4/SPI5/ I2S5	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3
Port B	PB0	-	TIM1_CH2N	TIM3_CH3	-	-	-	SPI5_SCK /I2S5_CK		-	-
	PB1	-	TIM1_CH3N	TIM3_CH4	-	-	-	SPI5_NSS /I2S5_WS		-	-
	PB2	-	-	-	-	-	-	-	-	-	-
	PB3	JTDO-SWO	TIM2_CH2	-	-	-	SPI1_SCK/I 2S1_CK	SPI3_SCK /I2S3_CK	USART1_ RX	-	I2C2_SDA
	PB4	JTRST		TIM3_CH1	-	-	SPI1_MISO	SPI3_MISO	I2S3ext_S D	-	I2C3_SDA
	PB5	-	-	TIM3_CH2	-	I2C1_SMB A	SPI1_MOSI /I2S1_SD	SPI3_MOSI/ I2S3_SD		-	-
	PB6	-	-	TIM4_CH1	-	I2C1_SCL	-	-	USART1_ TX	-	-
	PB7	-	-	TIM4_CH2	-	I2C1_SDA	-	-	USART1_ RX	-	-
	PB8	-	-	TIM4_CH3	TIM10_CH1	I2C1_SCL	-	SPI5_MOSI/ I2S5_SD	-	-	I2C3_SDA
	PB9	-	-	TIM4_CH4	TIM11_CH1	I2C1_SDA	SPI2_NSS/I 2S2_WS	-	-	-	I2C2_SDA

STEPS FOR UART TRANSMIT

//Enable clocK access to GPIOA

//SET PA2,PA3 MODE TO Alternate function Mode

//SET PA2 ,PA3 Alternative function type to UART_TX (AF07)

//CONFIGURE UART MODULE

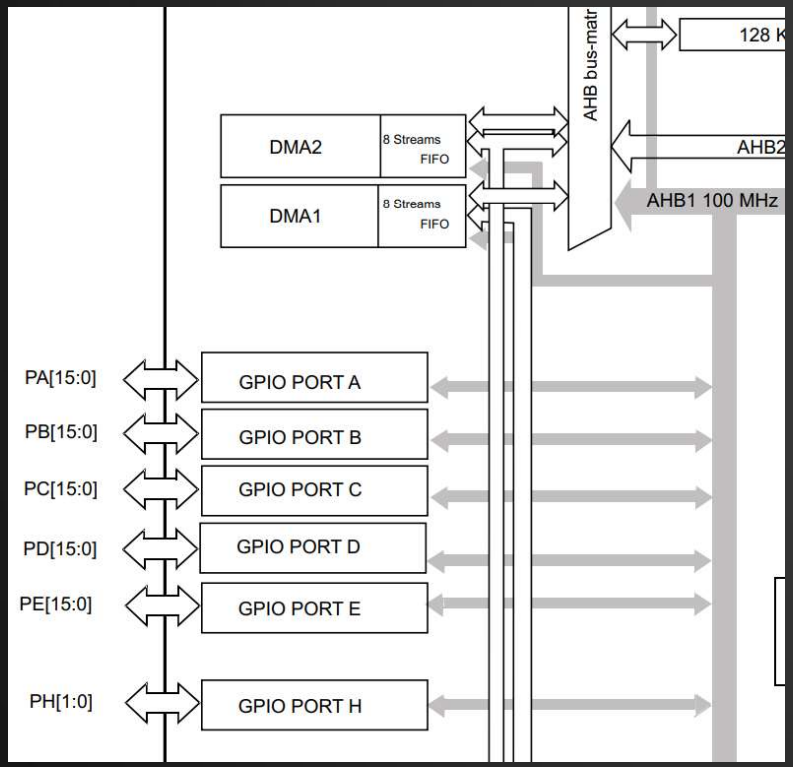
//ENABLE CLOCK ACCESS TO UART2

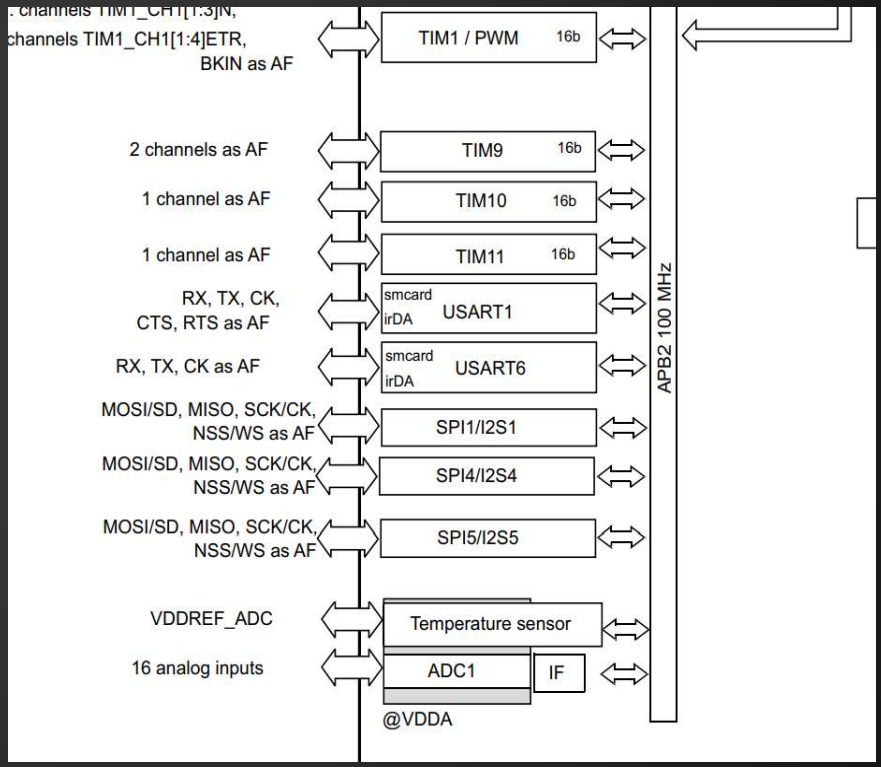
//CONFIGURE BAUDRATE

//CONFIGURE TRANSFER DIRECTION

//ENABLE UART MODULE

Block Diagram





USART1-FUNCTIONAL MAPPING

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	A
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/I2S1SPI2/ I2S2/SPI3/ I2S3	SPI2/I2S2/ SPI3/ I2S3/SPI4/ I2S4/SPI5/ I2S5	SPI3/I2S3/ USART1/ USART2	US
Port B	PB0	-	TIM1_CH2N	TIM3_CH3	-	-	-	SPI5_SCK /I2S5_CK		
	PB1	-	TIM1_CH3N	TIM3_CH4	-	-	-	SPI5_NSS /I2S5_WS		
	PB2	-	-	-	-	-	-	-	-	
	PB3	JTDO-SWO	TIM2_CH2	-	-	-	SPI1_SCK/I2S1_CK	SPI3_SCK /I2S3_CK	USART1_ RX	
	PB4	JTRST		TIM3_CH1	-	-	SPI1_MISO	SPI3_MISO	I2S3ext_SD	
	PB5	-		TIM3_CH2	-	I2C1_SMBA	SPI1_MOSI /I2S1_SD	SPI3_MOSI/ I2S3_SD		
	PB6	-	-	TIM4_CH1	-	I2C1_SCL	-	-	USART1_ TX	
	PB7	-	-	TIM4_CH2	-	I2C1_SDA	-	-	USART1_ RX	

FUNCTIONAL MAPPING

Table 9. Alternate function mapping

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/I2S1SPI2/ I2S2/SPI3/ I2S3	SPI2/I2S2/ SPI3/ I2S3/SPI4/ I2S4/SPI5/ I2S5	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3	OTG1_FS		SDIO			
Port A	PA0	-	TIM2_CH1/ TIM2_ETR	TIM5_CH1	-	-	-	-	USART2_ CTS	-	-	-	-	-	-	-	EVENT OUT
	PA1	-	TIM2_CH2	TIM5_CH2	-	-	SPI4_MOSI /I2S4_SD	-	USART2_ RTS	-	-	-	-	-	-	-	EVENT OUT
	PA2	-	TIM2_CH3	TIM5_CH3	TIM9_CH1	-	I2S2_CKIN	-	USART2_ TX	-	-	-	-	-	-	-	EVENT OUT
	PA3	-	TIM2_CH4	TIM5_CH4	TIM9_CH2	-	I2S2_MCK	-	USART2_ RX	-	-	-	-	-	-	-	EVENT OUT
	PA4	-	-	-	-	-	SPI1_NSS/I 2S1_WS	SPI3_NSS/I2 S3_WS	USART2_ CK	-	-	-	-	-	-	-	EVENT OUT
	PA5	-	TIM2_CH1/ TIM2_ETR	-	-	-	SPI1_SCK/I 2S1_CK	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA6	-	TIM1_BKIN	TIM3_CH1	-	-	SPI1_MISO	I2S2_MCK	-	-	-	-	-	SDIO_ CMD	-	-	EVENT OUT
	PA7	-	TIM1_CH1N	TIM3_CH2	-	-	SPI1_MOSI /I2S1_SD	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA8	MCO_1	TIM1_CH1	-	-	I2C3_ SCL	-	-	USART1_ CK	-	-	USB_FS_ SOF	-	SDIO_ D1	-	-	EVENT OUT
	PA9	-	TIM1_CH2	-	-	I2C3_ SMBA	-	-	USART1_ TX	-	-	USB_FS_ VBUS	-	SDIO_ D2	-	-	EVENT OUT

ALTERNATE FUNCTIONS

Address offset: 0x20
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
				0	1	1	1								

Bits 31:0 **AFRLy:** Alternate function selection for port x bit y (y = 0..7)
These bits are written by software to configure alternate function I/Os

AFRLy selection:

0000: AF0	1000: AF8
0001: AF1	1001: AF9
0010: AF2	1010: AF10
0011: AF3	1011: AF11
0100: AF4	1100: AF12
0101: AF5	1101: AF13
0110: AF6	1110: AF14
0111: AF7	1111: AF15

BAUD RATE CALCULATION

- Peripheral Clock =16000000
- BAUDRATE=9600
- BRR =((16000000+(9600/2))/9600)
- BRR= 1667
- HEX VALUE OF BRR =0X0683

UART WRITE

```
void uart2_write(int ch)
{
    //Make sure the transmit data register is empty
    while(!(*USART2_SR & 0x0080)){}
    //write to transmit data register
    *USART2_DR =(ch&0XFF);
}
```

TXE: Transmit data register empty

This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TXEIE bit =1 in the USART_CR1 register. It is cleared by a write to the USART_DR register.

0: Data is not transferred to the shift register

1: Data is transferred to the shift register)

26.6.1 Status register (USART_SR)

Address offset: 0x00

Reset value: 0x00C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

USART DATA REGISTER

Data register (USART_DR)

Address offset: 0x04

Reset value: 0xXXXX XXXX

Bits 31:9 Reserved, must be kept at reset value

Bits 8:0 **DR[8:0]**: Data value

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR)

The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 1).

The RDR register provides the parallel interface between the input shift register and the internal bus.

When transmitting with the parity enabled (PCE bit set to 1 in the USART_CR1 register), the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity.

When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

CONNECTION DETAILS

//PB6->TX
//PB7->RX ->BLUETOOTH TX
//RELAY -PA5

DEMO

Mindset Activity

- ✓ Write Down Your Top 10 Goals.(1 Mark)
- ✓ Write Down Your Top 10 Ideas to Achieve Your Goal.(1 Mark)
- ✓ 30 Minutes for Workout (5000-7000 Steps a Day)(2 Mark).
- ✓ 15 Minutes to Meditate (2 Mark)
- ✓ 10 Minutes to Visualize of Achieving Your Goals(1)
- ✓ 10 Minutes to Focus on the Day Plan (1)
- ✓ 2 Hr's for Learning and Take Notes. (2 Mark)

10/10

https://www.facebook.com/groups/embeddedsystemsandiot/

THANK YOU