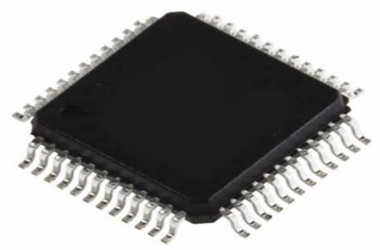
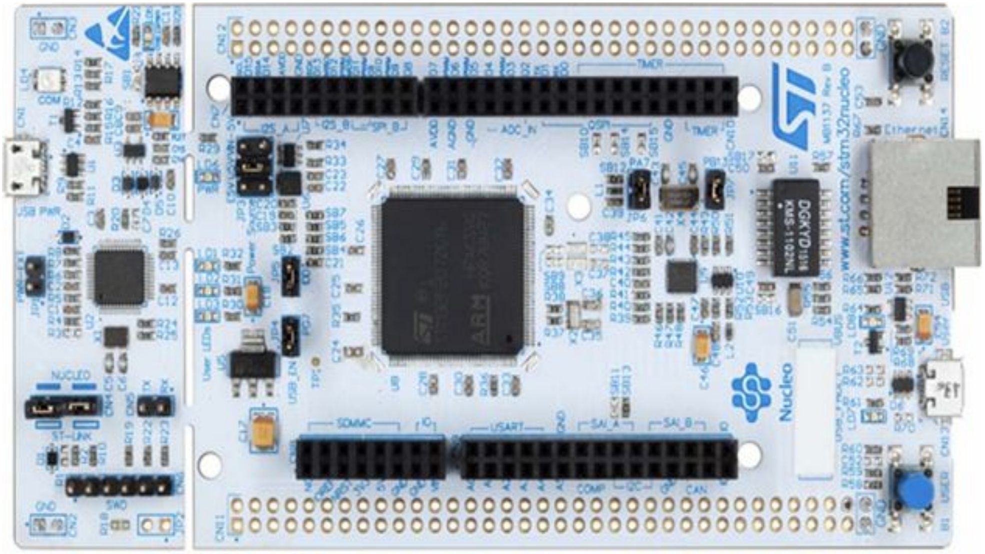
STM32 Microcontroller



Department of Computer Science and Engineering
BRAC University

STM32 Nucleo Board

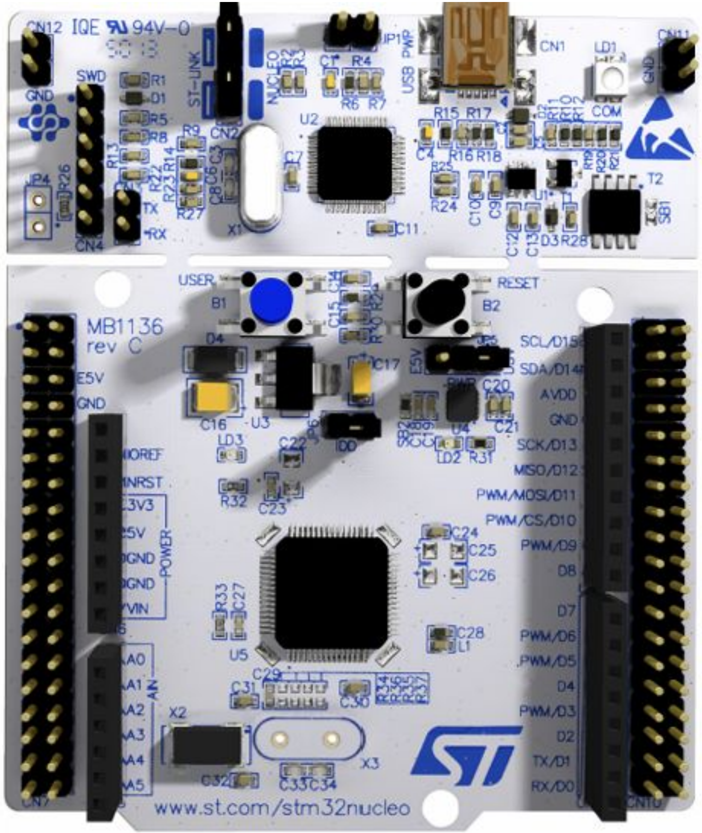
- 144 pin board



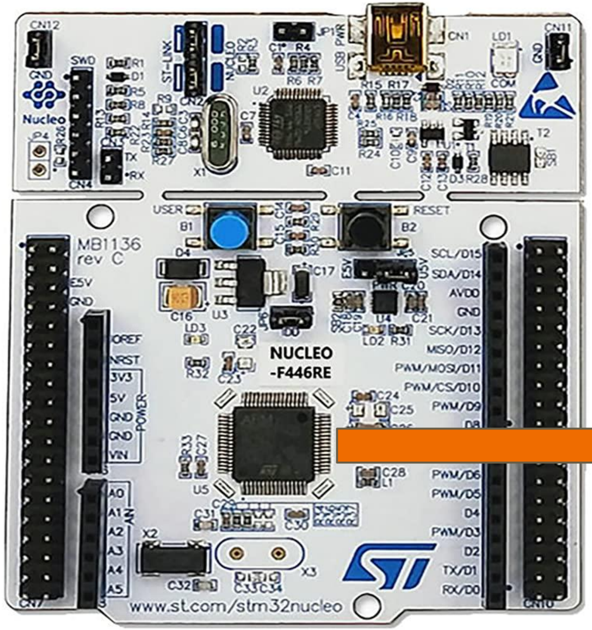
BRAC UNIVERSITY
Inspiring Excellence

STM32 Nucleo Board

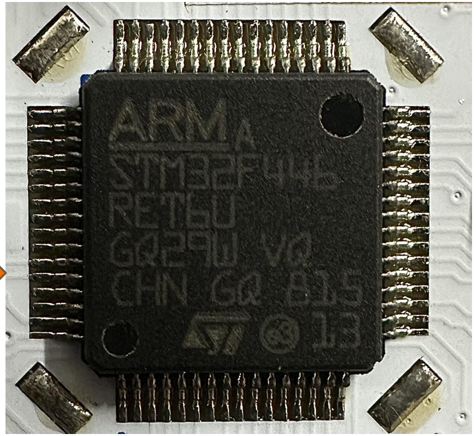
- 64 pin board



STM32F446RE



STM32F446RE Board



ARM Cortex-M4 Microcontroller

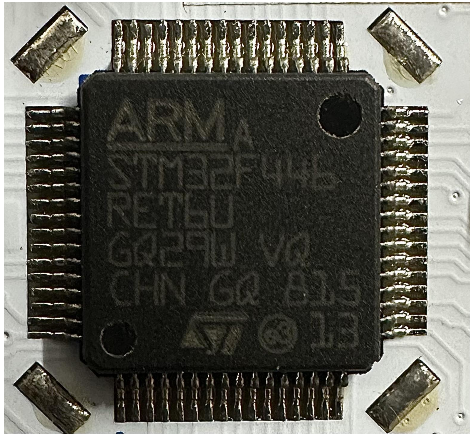
BRAC UNIVERSITY
Inspiring Excellence

Why ARM processor?

- Engineering
- IoT research
- Economy of Bangladesh
- Industry machinery
- Program almost all machine around us
- High level hardware (ARM processor building - Trillion $ market)

ARM Cortex-M4 Microcontroller

- 32 bit microprocessor with FPU [Floating Point Unit] core
- Flash Memory & RAM
- Peripheral port/devices



Word: data of 32-bit length.

Half-word: data of 16-bit length.

Byte: data of 8-bit length.

Double word: data of 64-bit length.

BRAC UNIVERSITY
Inspiring Excellence

Key Features of the ARM Cortex-M4 Core in STM32F446RE

1. Core Architecture

ARM Cortex-M4: A 32-bit RISC processor core designed for high performance and low power consumption.

DSP Instructions: The Cortex-M4 includes a set of digital signal processing (DSP) instructions that enhance its performance in applications requiring complex mathematical computations.

Floating Point Unit (FPU): Supports single-precision floating-point operations, which is beneficial for applications involving arithmetic operations with real numbers.

BRAC UNIVERSITY
Inspiring Excellence

Key Features of the ARM Cortex-M4 Core in STM32F446RE

2. Performance

Clock Speed: Can operate at up to 180 MHz, providing high processing power for complex applications.

Harvard Architecture: Features separate instruction and data buses, allowing simultaneous access to memory and instructions.

3. Memory

Flash Memory: 512 KB of on-chip Flash memory for program storage.

SRAM: 128 KB of SRAM for data storage.

Memory Protection Unit (MPU): Enhances security by allowing control over memory access permissions.

BRAC UNIVERSITY
Inspiring Excellence

Key Features of the ARM Cortex-M4 Core in STM32F446RE

4. Interrupt System

Nested Vectored Interrupt Controller (NVIC): Supports efficient handling of interrupts with 16 levels of priority.

SysTick Timer: A dedicated timer for generating system ticks, useful for real-time operating systems.

5. Peripherals

Timers: Multiple general-purpose timers, advanced-control timers, and basic timers for timing and counting applications.

Analog Peripherals: Includes ADCs (Analog-to-Digital Converters), DACs (Digital-to-Analog Converters), and comparators.

BRAC UNIVERSITY
Inspiring Excellence

Key Features of the ARM Cortex-M4 Core in STM32F446RE

7. Communication Interfaces:

UART/USART: For asynchronous and synchronous serial communication.

SPI: For synchronous serial communication with peripherals like sensors and memory devices.

I2C: For communication with low-speed peripherals.

CAN: For automotive and industrial communication networks.

USB: Supports both Host and Device modes.

GPIO: General-purpose input/output pins for interfacing with external components.

BRAC UNIVERSITY
Inspiring Excellence

Key Features of the ARM Cortex-M4 Core in STM32F446RE

8. Debug and Trace

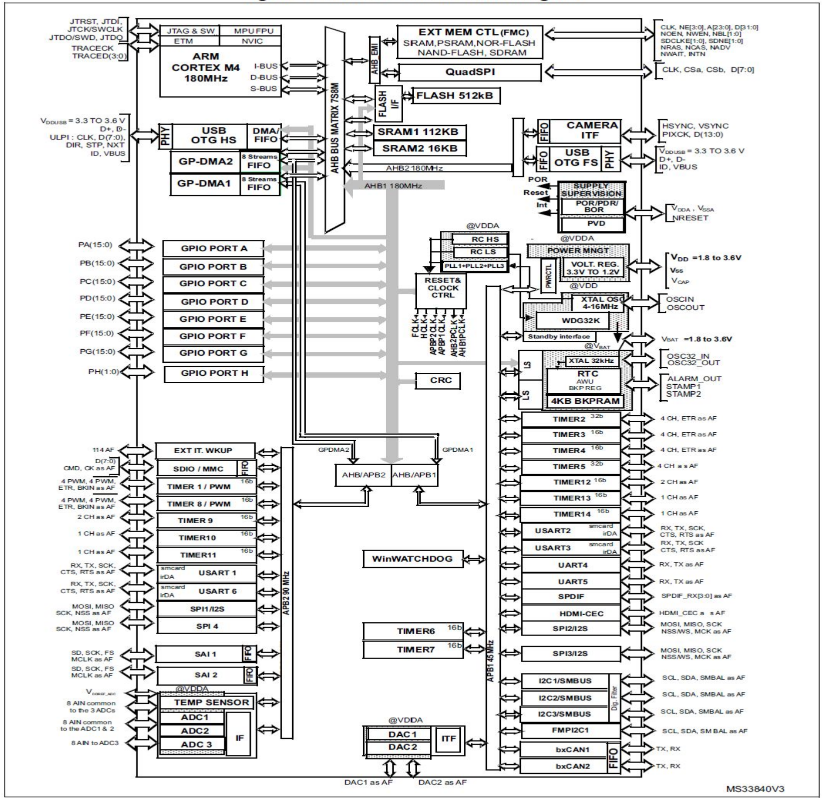
Embedded Trace Macrocell (ETM): Provides real-time trace capabilities for debugging complex applications.

Serial Wire Debug (SWD): A two-pin protocol for debugging, offering a reduced pin count compared to JTAG.

BRAC UNIVERSITY
Inspiring Excellence

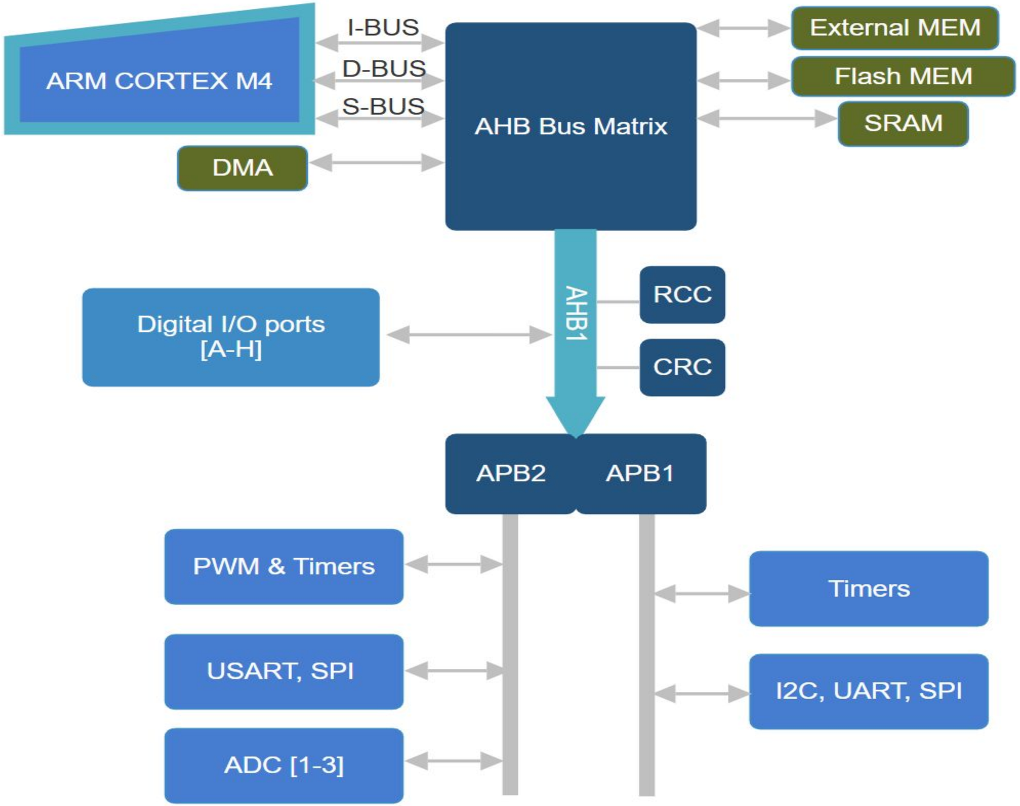
STM32F446RE Block Diagram of the System Architecture

Figure 3. STM32F446xC/E block diagram



For better view click on [this](#).

System Architecture [Simplified - Only AHB1]



BRAC UNIVERSITY
Inspiring Excellence

Memory & Bus Architecture

- **I-bus:** This bus connects the Instruction bus of the Cortex M4 with FPU core to the Bus Matrix. This bus is used by the core to fetch instructions.

- **D-bus:** This bus connects the databus of the Cortex M4 with FPU core to the Bus Matrix. This bus is used by the core for literal load and debug access. The target of this bus is a memory containing code or data.

- **S-bus:** This bus connects the system bus of the Cortex M4 with FPU core to the Bus Matrix. This bus is used to access data located in a peripheral or in SRAM. Instructions may also be fetch on this bus (less efficient than I-bus).

Memory & Bus Architecture

- **DMA, Flash MEM, External MEM, SRAM**
- **AHB Bus Matrix**
- **APB Buses:** The two APB bridges, APB1 and APB2, provide full synchronous connections between the AHB and the two APB buses, allowing flexible selection of the peripheral frequency. After each device reset, all peripheral clocks are disabled (except for the SRAM and Flash memory interface). Before using a peripheral you have to enable its clock in the RCC_AHBxENR or RCC_APBxENR register.

CRC & RCC

Cyclic Redundancy Check (CRC)

The CRC calculation unit is used to get a CRC code from a 32-bit data word and a fixed generator polynomial. Among other applications, CRC-based techniques are used to verify data transmission or storage integrity.

Reset and Clock Control (RCC)

There are three types of reset

1. System Reset
2. Power Reset
3. Backup domain Reset

RCC AHB Peripheral Clock Enable Register [RCC->AHB1ENR]

- 32 bit register
- Can enable port any digital I/O port [A-H]
- Can enable DMAs
- Can enable CRC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OTGHS ULPIEN	OTGHS EN	Res.	Res.	Res.	Res.	Res.	Res.	DMA2 EN	DMA1 EN	Res.	Res.	BKP SRAMEN	Res.	Res.
	rw	rw							rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC EN	Res.	Res.	Res.	Res.	GPIOH EN	GPIOG EN	GPIOF EN	GPIOE EN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw					rw	rw	rw	rw	rw	rw	rw	rw

BRAC UNIVERSITY
Inspiring Excellence

RCC AHB Peripheral Clock Enable Register [RCC->AHB1ENR]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OTGHS ULPIEN	OTGHS EN	Res.	Res.	Res.	Res.	Res.	Res.	DMA2 EN	DMA1 EN	Res.	Res.	BKP SRAMEN	Res.	Res.
	rw	rw							rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC EN	Res.	Res.	Res.	Res.	GPIOH EN	GPIOG EN	GPIOF EN	GPIOE EN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw					rw	rw	rw	rw	rw	rw	rw	rw

BIT 0 - GPIOA EN: I/O port A clock enable

0: IO port A clock disabled

1: IO port A clock enabled

BIT 1 - GPIOB EN: I/O port B clock enable

0: IO port A clock disabled

1: IO port A clock enabled

.... and so on

Example:

```
RCC->AHB1ENR |= 1<<0
```

						1
5	4	....	3	2	1	0

```
RCC->AHB1ENR |= 1<<4
```

			1	0	0	0
5	4	....	3	2	1	0

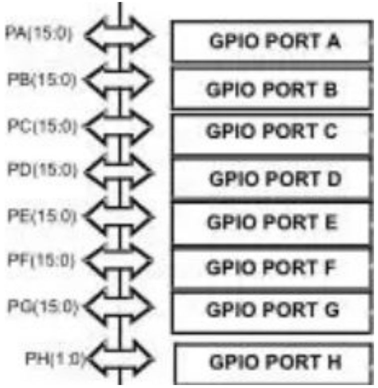
BRAC UNIVERSITY

Inspiring Excellence

General Purpose I/O [GPIO]

Each general-purpose I/O port has -

- A four 32-bit **configuration** registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR),
- Two 32-bit **data** registers (GPIOx_IDR and GPIOx_ODR),
- A 32-bit **set/reset** register (GPIOx_BSRR),
- A 32-bit **locking** register (GPIOx_LCKR)
- Two 32-bit **alternate function selection** register (GPIOx_AFRH and GPIOx_AFRL).



BRAC UNIVERSITY
Inspiring Excellence

Main features of GPIO

- Up to 16 I/Os under control
- Output states: push-pull or open drain + pull-up/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input states: floating, pull-up/down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Bit set and reset register (GPIOx_BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O configuration
- Analog function
- Alternate function input/output selection registers (at most 16 AFs per I/O)
- Fast toggle capable of changing every two clock cycles
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions

GPIO functional description

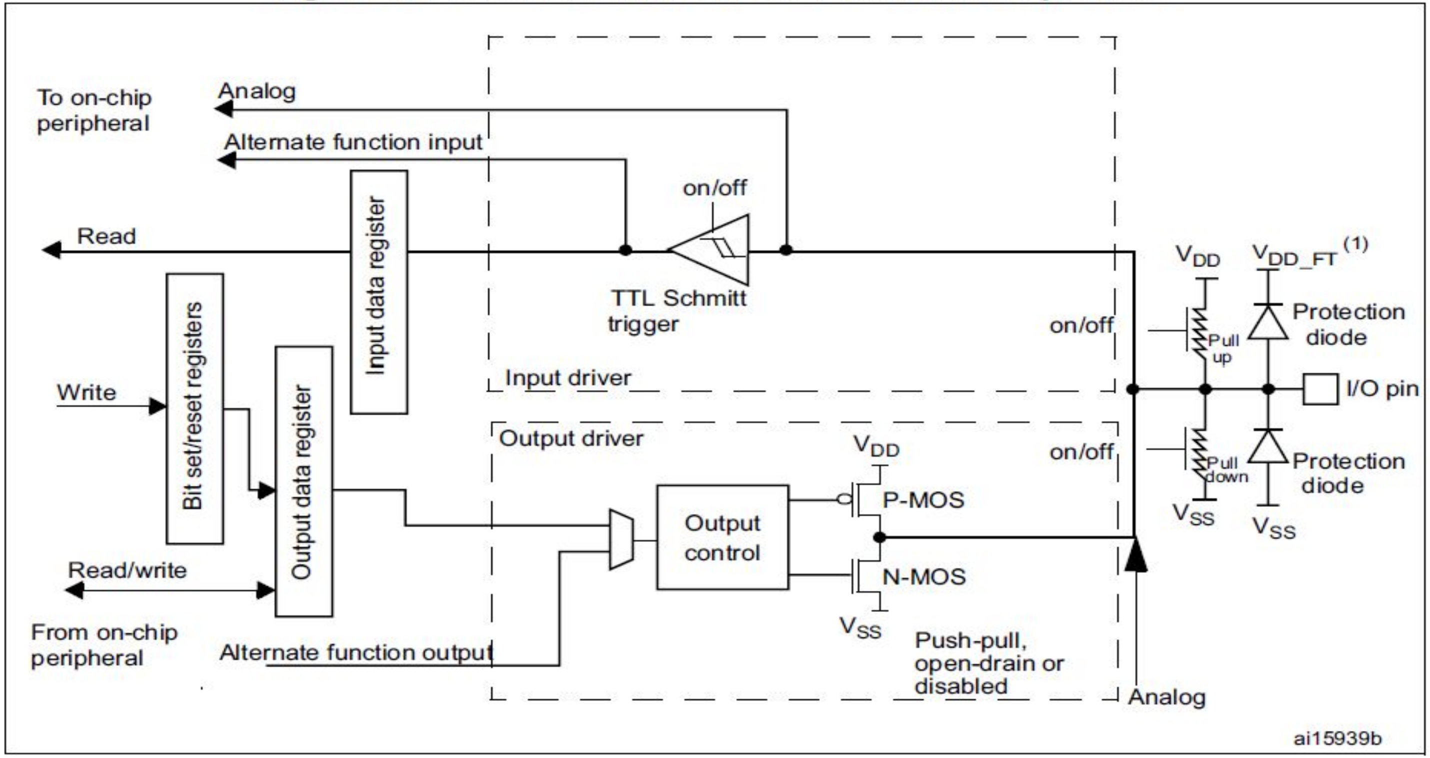
Subject to the specific hardware characteristics of each I/O port listed in the datasheet, each port bit of the general-purpose I/O (GPIO) ports can be individually configured by software in several modes:

- Input floating
- Input pull-up
- Input-pull-down
- Analog
- Output open-drain with pull-up or pull-down capability
- Output push-pull with pull-up or pull-down capability
- Alternate function push-pull with pull-up or pull-down capability
- Alternate function open-drain with pull-up or pull-down capability
- Each I/O port bit is freely programmable, however the I/O port registers have to be
- accessed as 32-bit words, half-words or bytes. The purpose of the GPIOx_BSRR register is
- to allow atomic read/modify accesses to any of the GPIO registers. In this way, there is no
- risk of an IRQ occurring between the read and the modify access.

BRAC UNIVERSITY
Inspiring Excellence

Internal Interfacing Unit of each GPIO PORT

Figure 18. Basic structure of a 5 V tolerant I/O port bit



BRAC UNIVERSITY

Inspiring Excellence

1. GPIO Configuration Registers

GPIOx_MODER (x = A..H) : These bits configure the I/O direction mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
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
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit [0..31]: MODERy[1:0]: Port x configuration bits (y = 0..15)

00: Input (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

1. GPIO Configuration Registers

GPIOx_OTYPER (x=A..H): These bits are written by software to configure the output type of the I/O port.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 - Reserved
Bits 15:0 - OTy: Port x configuration bits (y = 0..15)

0: Output push-pull (reset state)
1: Output open-drain

BRAC UNIVERSITY
Inspiring Excellence

1. GPIO Configuration Registers

GPIOx_OSPEEDR (x=A..H): These bits are to configure the I/O output speed.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
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
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3[1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

OSPEEDRy[1:0]: Port x configuration bits (y = 0..15)

00: Low speed
01: Medium speed
10: Fast speed
11: High speed

1. GPIO Configuration Registers

GPIOx_PUPDR (x=A..H): These bits are written to configure the I/O pull-up or pull-down

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
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**
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits[0..31] - PUPDRy[1:0]: Port x configuration bits (y = 0..15)

00: No pull-up, pull-down
01: Pull-up
10: Pull-down
11: Reserved

BRAC UNIVERSITY
Inspiring Excellence

2. Data Registers

GPIOx_IDR (x=A..H): These bits are read-only. They contain the input value of the corresponding I/O port.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 - Reserved

Bits 15:0 - IDRy: Port input data (y = 0..15)

BRAC UNIVERSITY

Inspiring Excellence

2. Data Registers

GPIOx_ODR (x=A..H): These bits can be read and written

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 - Reserved

Bits 15:0 - ODRy: Port output data (y = 0..15)

Note: For atomic bit set/reset, the ODR bits can be individually set and reset by writing to the GPIOx_BSRR register (x = A..H).

BRAC UNIVERSITY
Inspiring Excellence

3. Set/Rest Register

GPIOx_BSRR (x=A..H): These bits are write-only and can be accessed in word, half-word or byte mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 - BRy: Port x reset bit y (y = 0..15)

0: No action on the corresponding ODRx bit
1: Resets the corresponding ODRx bit

Bits 15:0 - BSy: Port x set bit y (y= 0..15)

0: No action on the corresponding ODRx bit
1: Sets the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

BRAC UNIVERSITY
Inspiring Excellence

THANK YOU!

BRAC UNIVERSITY
Inspiring Excellence