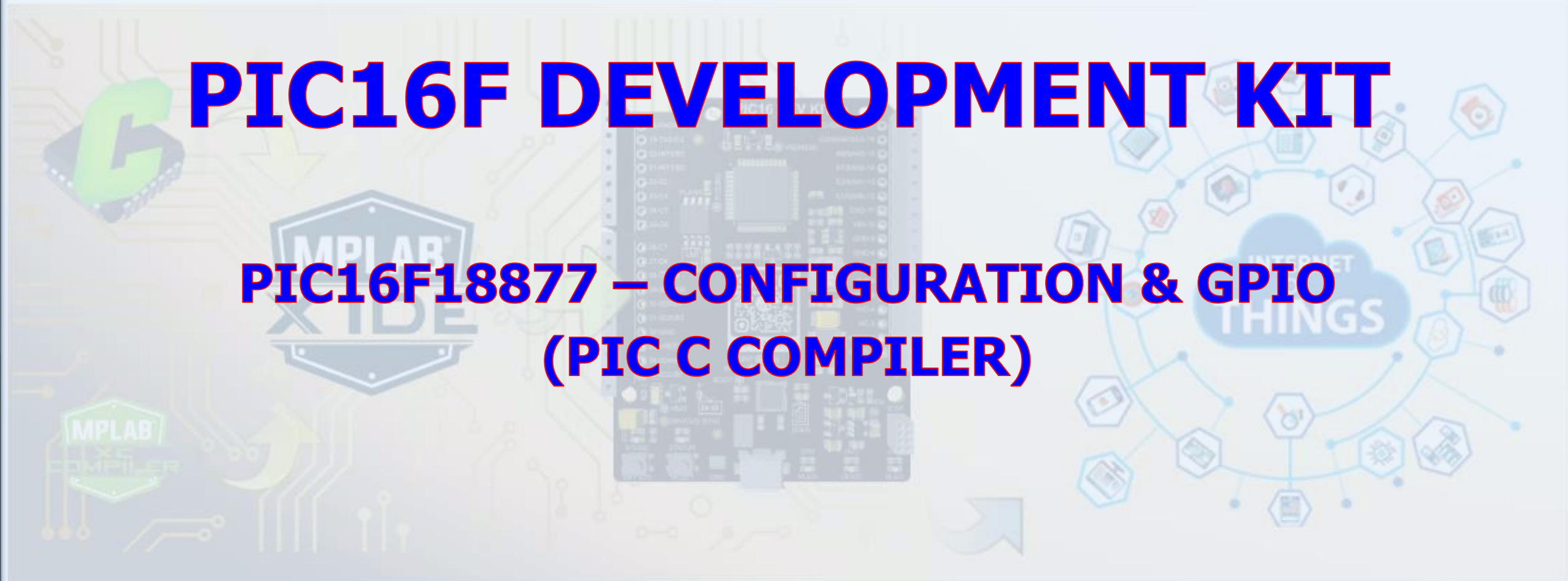


PIC16F DEVELOPMENT KIT

**PIC16F18877 – CONFIGURATION & GPIO
(PIC C COMPILER)**





NỘI DUNG

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT



GIỚI THIỆU CHUNG

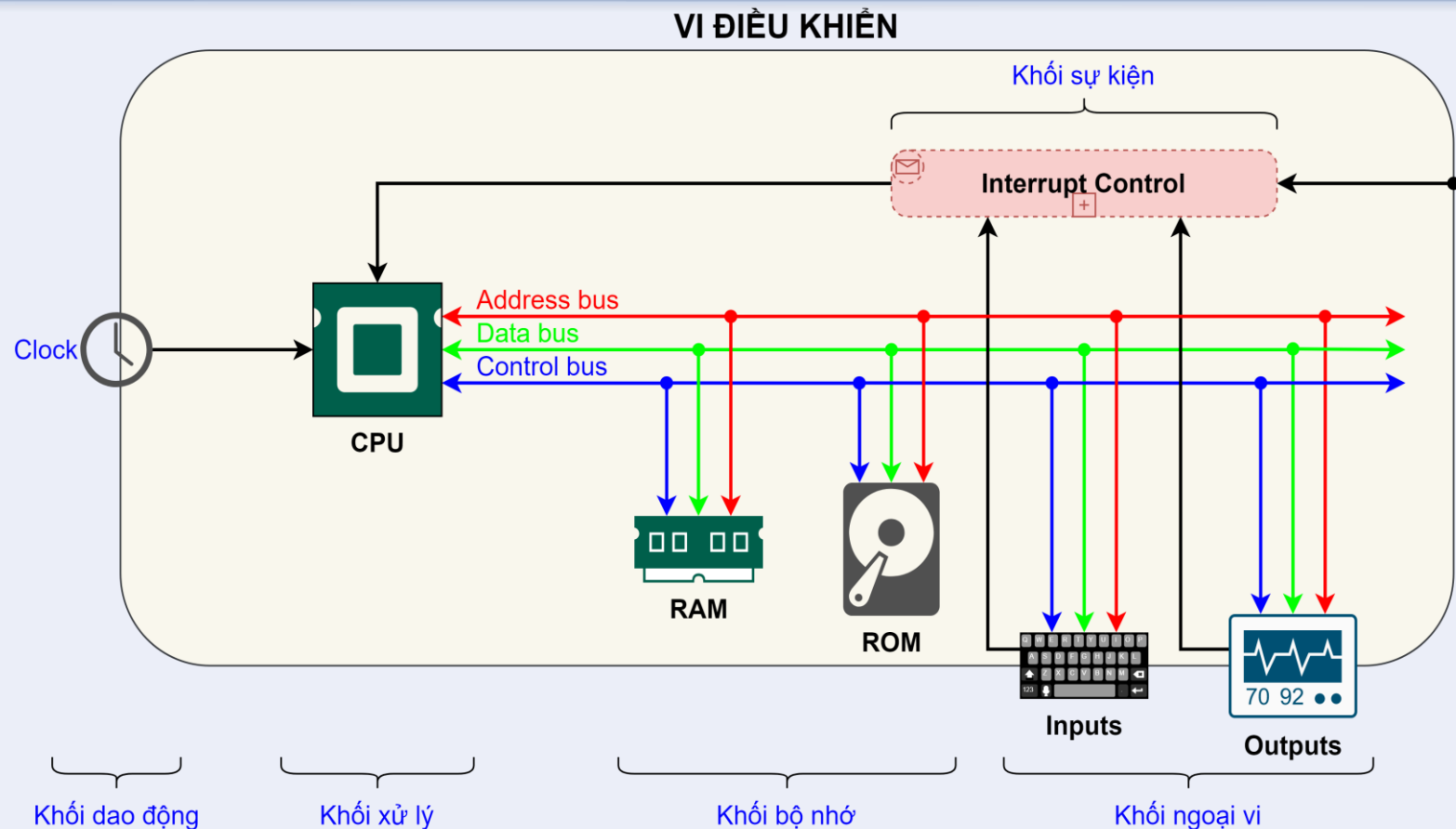
1. HỆ THỐNG VI ĐIỀU KHIỂN

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

1. Khối dao động
2. Khối xử lý
3. Khối bộ nhớ
4. Khối ngoại vi
5. Khối sự kiện
6. Các khối khác:
Quản lý nguồn,
Gỡ lỗi, ...





GIỚI THIỆU CHUNG

2. CÔNG CỤ PHÁT TRIỂN

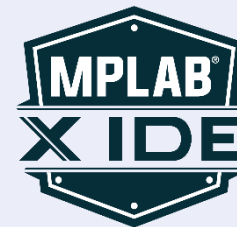
GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

1. MPLAB X IDE/ PIC C IDE

Môi trường soạn thảo, định dạng, gỡ lỗi, ...

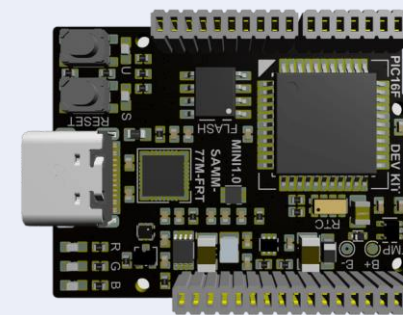


2. MPLAB XC8/ CCS C

Chuyển đổi ngôn ngữ C/Asm sang mã thực thi.



3. Evaluation Kit





GIỚI THIỆU CHUNG

3. TÀI LIỆU KỸ THUẬT

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

1. Datasheet: [DS40001825F](#)
2. User's guide: [C:\Program Files \(x86\)\PICC\ccsc.chm](#)
3. Code Examples: [C:\Program Files \(x86\)\PICC\Examples](#)



CONFIGURATION

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

Để làm việc với vi điều khiển cần:

- ☐ Cấu hình được chip.
- ☐ Quản lý được bộ nhớ.
- ☐ Quản lý được ngoại vi.



CONFIGURATION

1. CẤU HÌNH DAO ĐỘNG

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

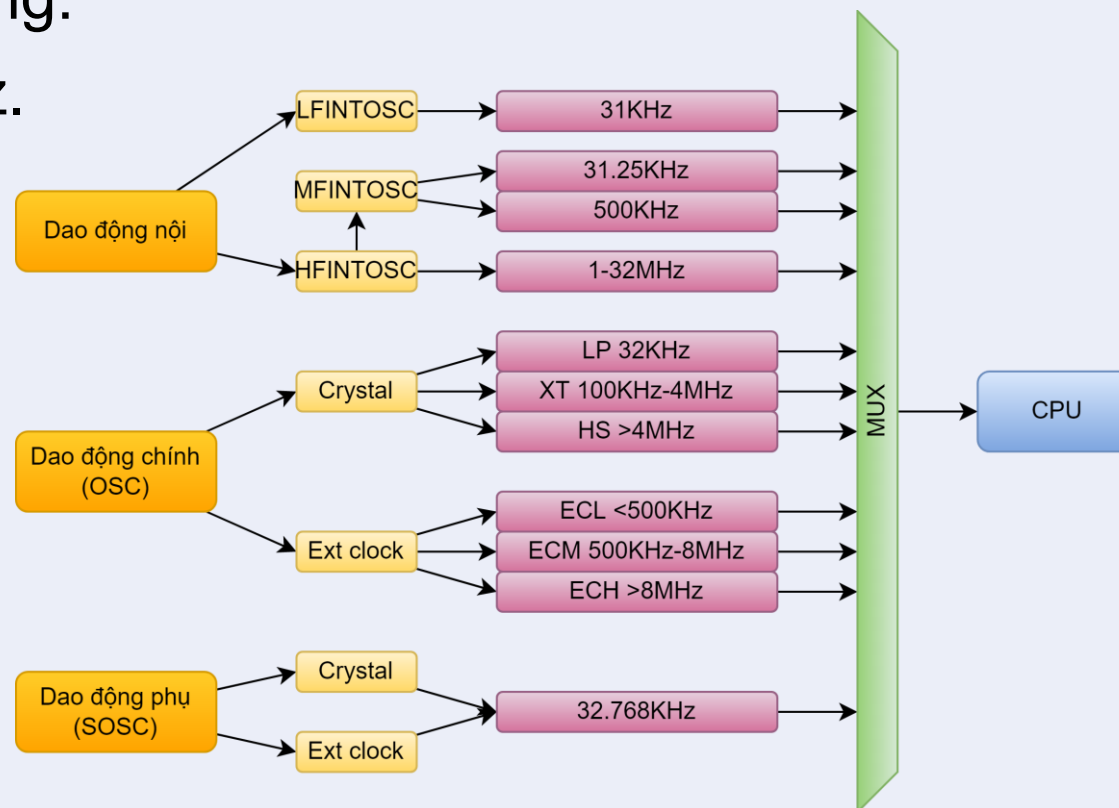
❑ PIC16F18877 có 2 nhóm dao động:

➤ Dao động nội: 31KHz đến 32MHz.

➤ Dao động ngoài:

- Clock chính OSC: lên đến 32MHz.
- Clock phụ SOSC: 32768Hz.

❑ Tốc độ xử lý: $F_{CPU} = F_{OSC} / 4$.





CONFIGURATION

1. CẤU HÌNH DAO ĐỘNG

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

- Kit sử dụng dao động nội.
- Tốc độ tối đa 32MHz.

Project Wizard - F:\Git\SAMPI\P16DK\P16F1887x\TUT\B02_Config_GPIO\B02_1_CCS\main.ccsproj

File Help

Options Code

Device Selection

Device

Example Code

Peripherals

Analog

Communications

SPI

Drivers

Header Files

High/Low Voltage

Interrupts

I/O Pins

Timers 0-2

Timers 3-up

LCD (Internal)

General

Device

Family: PIC16

Device: PIC16F18877

☐ Debug Code

Fixed Compiler Version: None

Clock

Oscillator Type: Internal

Clock Speed: 32 MHz 8 MIPS

CPU Clock Speed: 20 MHz

☐ Use USB Low Speed

☐ Use USB Full Speed

☐ Clock Out

WDT

☐ Enabled

Check any of the following to restart WDT during calls to:

☐ getc() and fgetc()

☐ I2c_read()

☐ Delay Functions

Reset: 1.0 ms

WDT Reset

☒ 1 ms ☐ 1024 ms

☐ 2 ms ☐ 2048 ms

☐ 4 ms ☐ 4096 ms

☐ 8 ms ☐ 8192 ms

☐ 16 ms ☐ 16384 ms

☐ 32 ms ☐ 32768 ms

☐ 64 ms ☐ 65536 ms

☐ 128 ms ☐ 131072 ms

☐ 256 ms ☐ 262144 ms

☐ 512 ms

Chip: PIC16F18877 Frequency: 32,000,000

Create Project Cancel



CONFIGURATION

2. CẤU HÌNH KHÁC

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

- Có thể sử dụng cấu hình mặc định.

Project Wizard - F:\Git\SAMPI\P16DK\P16F1887x\TUT\B02_Config_GPIO\B02_1_CCS\main.ccsproj

File Help

Options Code

Fuses

☐ Include Fuses

☐ External Oscillator not enabled
On Power-up clock running from External Oscillator

☐ Output clock on OSC2

☒ Clock Switching Enabled

☒ Fail-safe clock monitor enabled

☒ Master Clear pin enabled

☐ Power Up Timer

☐ Low-Power Brownout reset is enabled
No brownout reset

Brownout reset at 2.4V

☒ Zero-cross detect circuit is disabled at POR

☒ Allows only one reconfiguration of peripheral pins

☒ Stack full/underflow will cause reset

Watch Dog Timer Postscale settable in software

Watchdog Window is settable in software

WDT clock source settable in software

Program memory not write protected

☒ Scanner module is available for use

☒ Low Voltage Programming on B3(PIC16) or B5(PIC18)

Chip: PIC16F18877 Frequency: 32,000,000

Create Project Cancel



GPIO

1. I/O PORTS

GIỚI THIỆU CHUNG

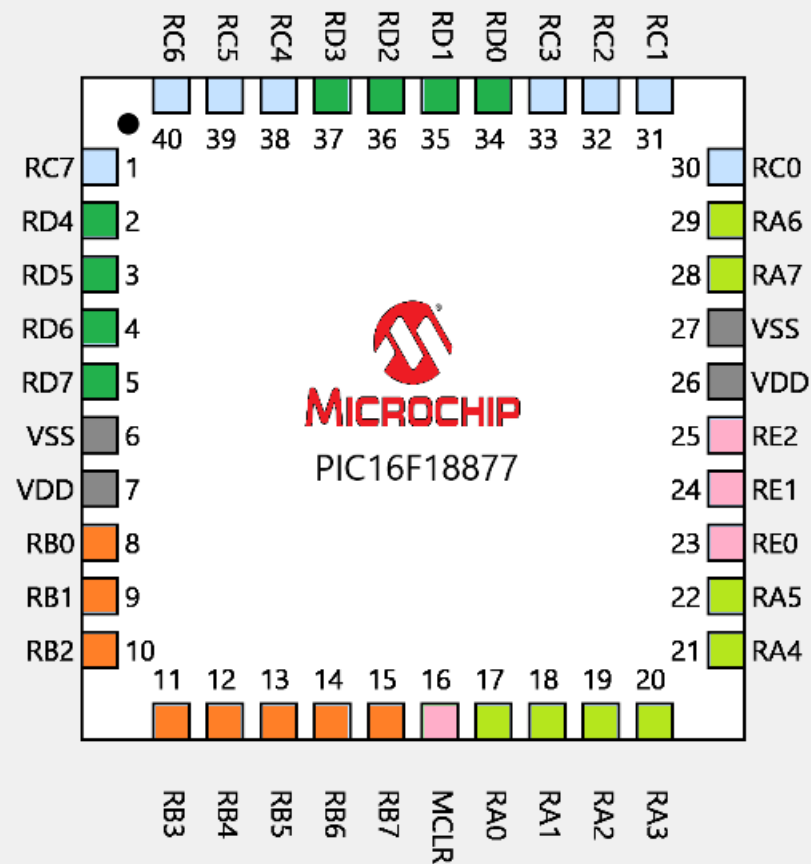
CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

❑ PIC16F18877 có 5 port:

Port A, B, C, D: 8 I/O.

Port E: 4 I/O. Có 1 I/O được mặc định chức năng MCLR.





GPIO

2. DIGITAL OUTPUTS

[GIỚI THIỆU CHUNG](#)[CONFIGURATION](#)[GENERAL-PURPOSE INPUT/OUTPUT](#)

❖ Cấu hình trong Project Wizard:

❖ Cấu hình trong code:

```
#use FIXED_IO(<P>_outputs=PIN_<P>1)
```

```
#use FIXED_IO(A_outputs=PIN_A1, PIN_A2)
```

Cấu hình ngõ ra:

```
set_tris_<P>(value);
```

Bit tương ứng có giá trị 0 thì pin sẽ là ngõ ra.

Đặt mức logic theo port:

```
output_<P>(value);
```

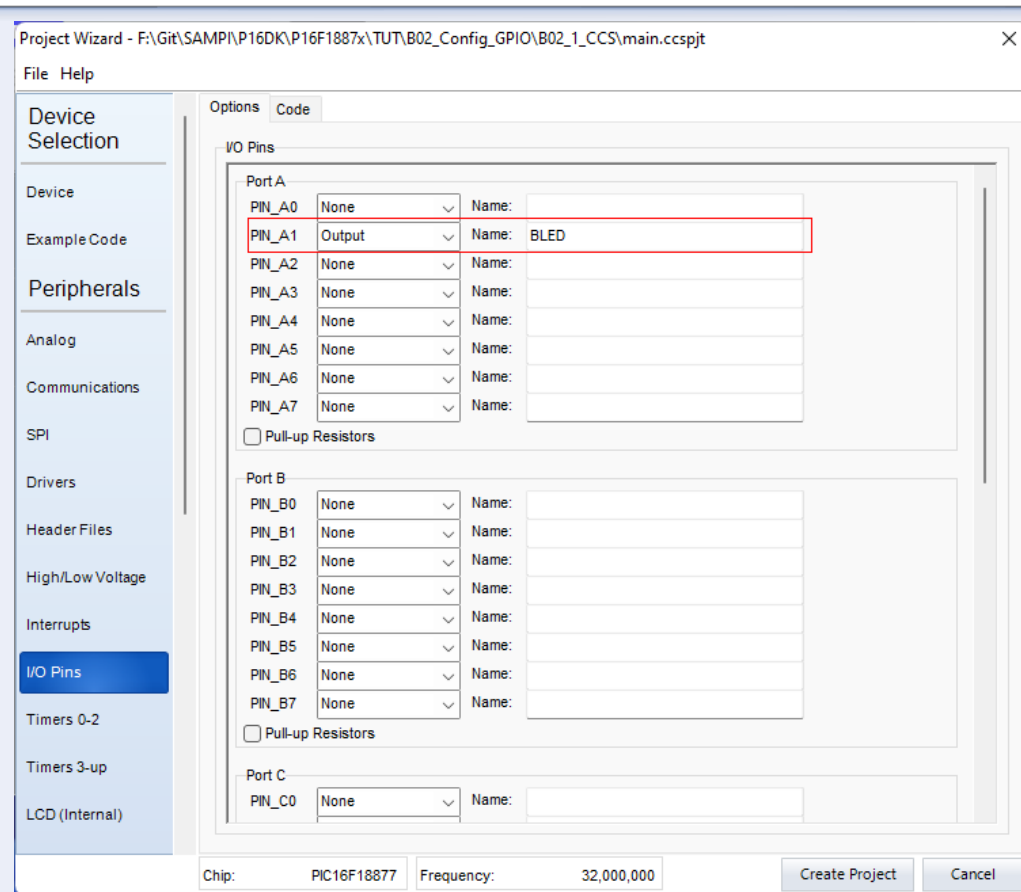
Đặt mức logic theo pin:

```
output_bit(PIN, logic);
```

```
output_high(PIN);
```

```
output_low(PIN);
```

```
output_toggle(PIN);
```





GPIO

3. DIGITAL INPUTS

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

❖ Cấu hình trong Project Wizard:

❖ Cấu hình trong code:

Cấu hình ngõ vào:

```
set_tris_<P>(value);
```

Bit tương ứng có giá trị 1 thì pin sẽ là ngõ vào.

Cấu hình pull-up:

```
port_<P>_pullups(value);
```

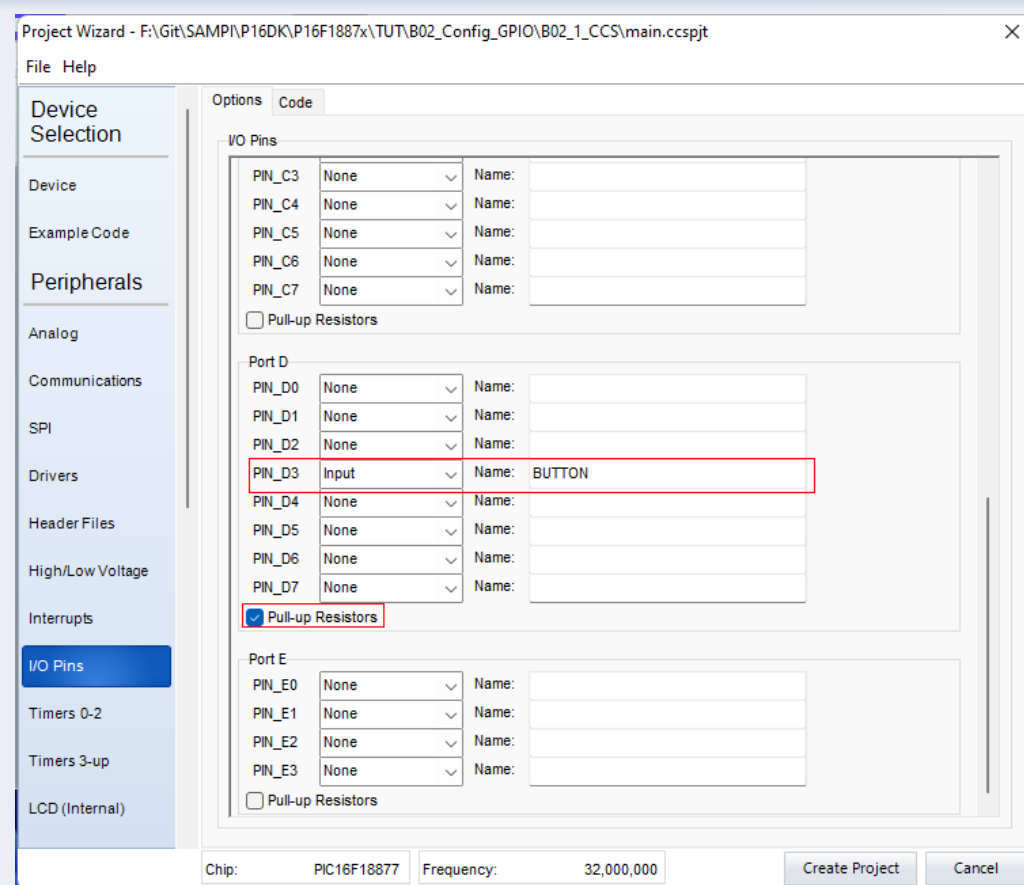
Bit tương ứng có giá trị 1 thì pin sẽ pull-up.

Đọc mức logic theo port:

```
logic=input_<P>(value);
```

Đọc mức logic theo pin:

```
Logic=input(PIN);
```





GPIO

3. VÍ DỤ

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

```
1  #include <main.h>
2
3  void main()
4  {
5      // A1 output
6      set_tris_a(0b11111101);
7      output_low(PIN_A1);
8      // D3 input
9      set_tris_d(0b11111111);
10     port_d_pullups(0b00001000);
11
12     while(TRUE)
13     {
14         output_toggle(PIN_A1);
15         delay_ms(100);
16         //output_bit(PIN_A1, !input(PIN_D3));
17     }
18 }
```

```
1  #include <main.h>
2
3  void main()
4  {
5      // A1 output
6      set_tris_a(0b11111101);
7      output_low(PIN_A1);
8      // D3 input
9      set_tris_d(0b11111111);
10     port_d_pullups(0b00001000);
11
12     while(TRUE)
13     {
14         //output_toggle(PIN_A1);
15         //delay_ms(100);
16         output_bit(PIN_A1, !input(PIN_D3));
17     }
18 }
```



GPIO

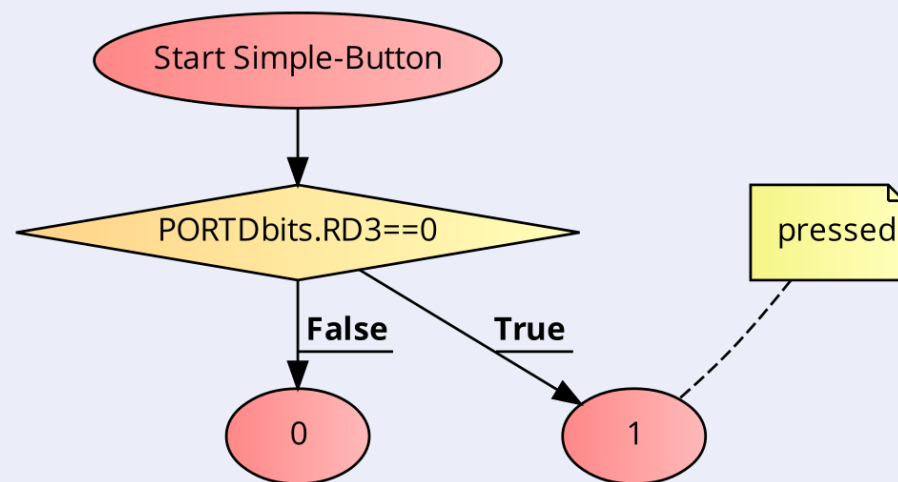
3. VÍ DỤ

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

Simple button:





GPIO

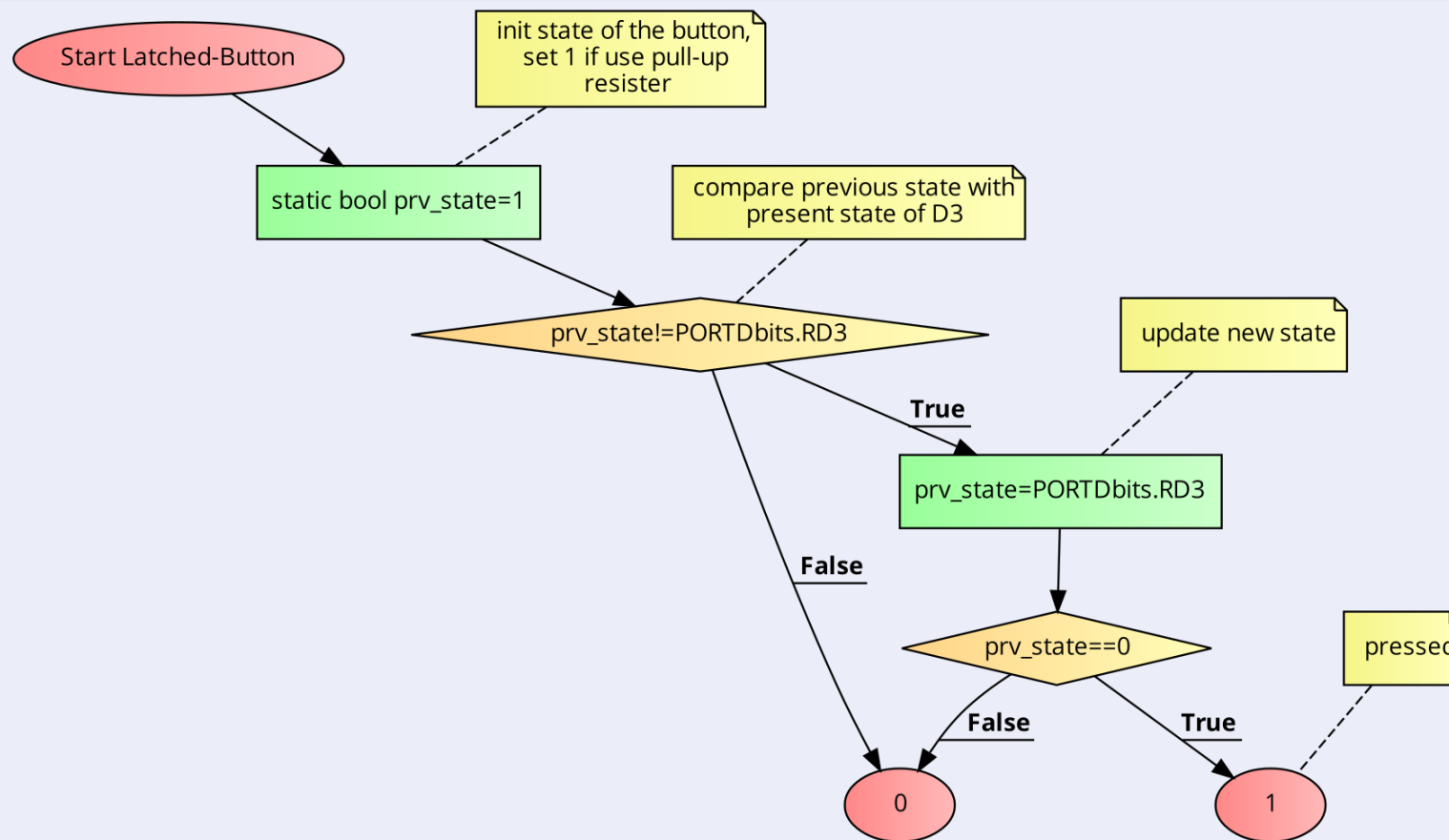
3. VÍ DỤ

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

Latched button:





GPIO

3. VÍ DỤ

GIỚI THIỆU CHUNG

CONFIGURATION

GENERAL-PURPOSE INPUT/OUTPUT

Debounce button:

