

Masking

Blocking out stuff we're not interested in

Reading 4 switches attached to PORTB

Atmega168 Pin Mapping

Arduino function

reset

digital pin 0 (RX)

digital pin 1 (TX)

digital pin 2

digital pin 3 (PWM)

digital pin 4

VCC

GND

crystal

crystal

digital pin 5 (PWM)

digital pin 6 (PWM)

digital pin 7

digital pin 8

(PCINT14/RESET) PC6

(PCINT16/RXD) PD0

(PCINT17/TXD) PD1

(PCINT18/INT0) PD2

(PCINT19/OC2B/INT1) PD3

(PCINT20/XCK/T0) PD4

VCC

GND

(PCINT6/XTAL1/TOSC1) PB6

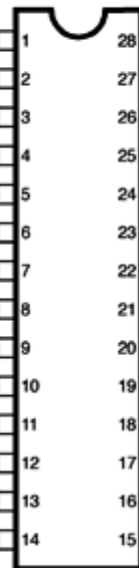
(PCINT7/XTAL2/TOSC2) PB7

(PCINT21/OC0B/T1) PD5

(PCINT22/OC0A/AIN0) PD6

(PCINT23/AIN1) PD7

(PCINT0/CLKO/ICP1) PB0



28 PC5 (ADC5/SCL/PCINT13)

27 PC4 (ADC4/SDA/PCINT12)

26 PC3 (ADC3/PCINT11)

25 PC2 (ADC2/PCINT10)

24 PC1 (ADC1/PCINT9)

23 PC0 (ADC0/PCINT8)

22 GND

21 AREF

20 AVCC

19 PB5 (SCK/PCINT5)

18 PB4 (MISO/PCINT4)

17 PB3 (MOSI/OC2A/PCINT3)

16 PB2 (SS/OC1B/PCINT2)

15 PB1 (OC1A/PCINT1)

Arduino function

analog input 5

analog input 4

analog input 3

analog input 2

analog input 1

analog input 0

GND

analog reference

VCC

digital pin 13

digital pin 12

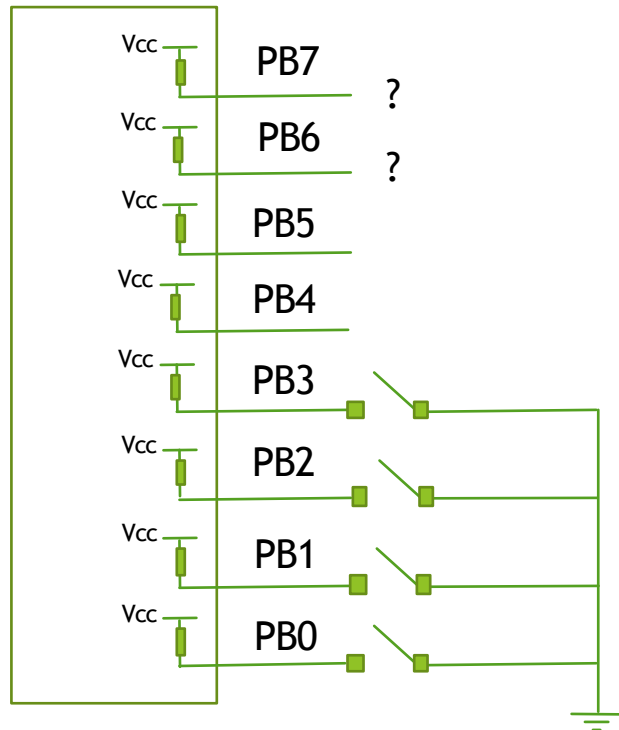
digital pin 11 (PWM)

digital pin 10 (PWM)

digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

What is in PINB?



DDRB = 0x00;

DDRB

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

PORTB = 0xFF;

PORTB

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

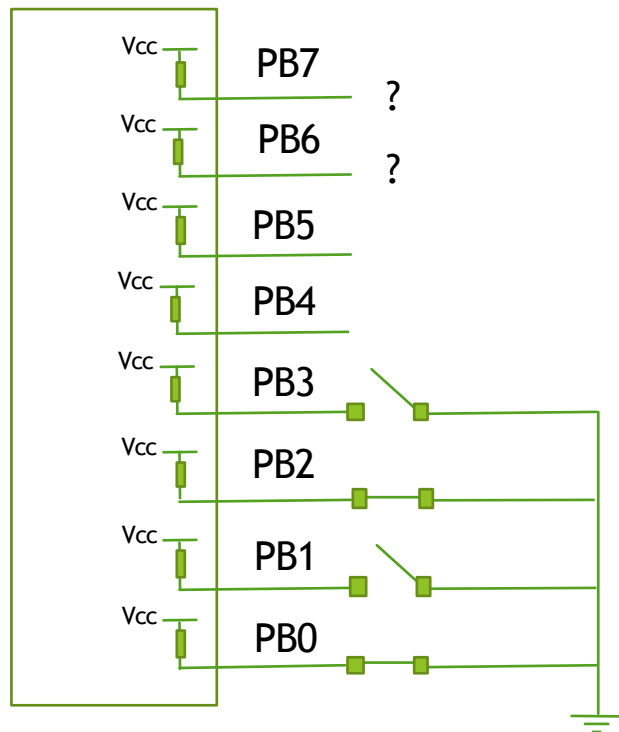
PINB

?	?	1	1	1	1	1	1
---	---	---	---	---	---	---	---

if(PINB == 0xFF) //?????

if(PINB == 0x3F) //?????

What is in PINB?



DDRB = 0x00;

DDRB

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

PORTB = 0xFF;

PORTB

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

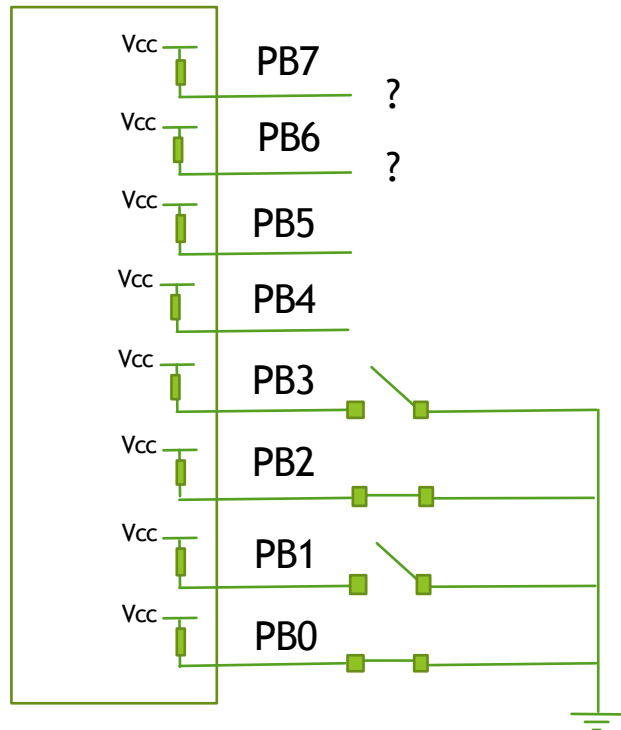
PINB

?	?	1	1	1	0	1	0
---	---	---	---	---	---	---	---

if(PINB == 0xFA) //?????

if(PINB == 0x3A) //?????

Better to “mask out” bits not interested in



PINB	?	?	1	1	1	0	1	0
------	---	---	---	---	---	---	---	---

```
x = PINB;
```

x	?	?	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---

```
x = x & 0x0F;
```

mask

Diagram illustrating the AND operation for bit masking:

- Input 1 (x):

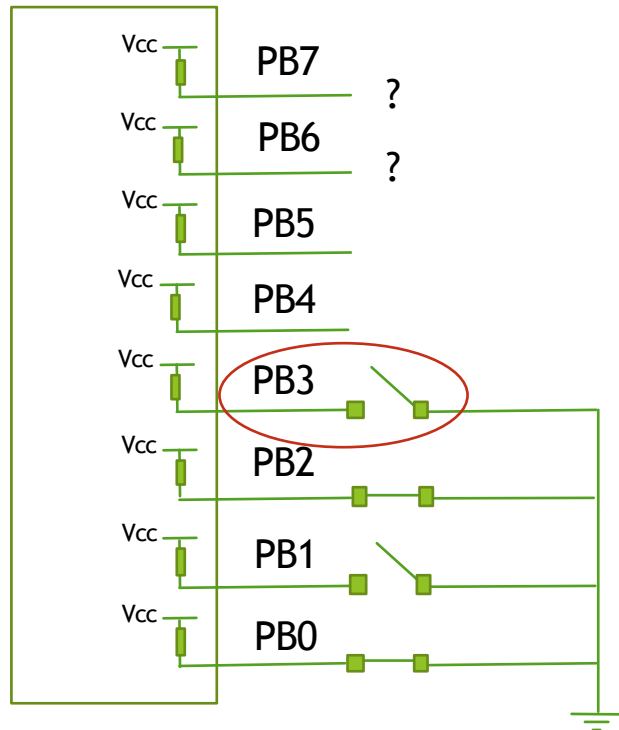
?	?	1	1	1	0	1	0
---	---	---	---	---	---	---	---
- Input 2 (mask):

0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---
- Operation: $\&$ (AND)
- Result (x):

0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---

```
if(x == 0x0A)
{
    ...
}
```

What if we only want to read one switch?



PINB

?	?	1	1	1	0	1	0
---	---	---	---	---	---	---	---

x = PINB;

x

?	?	1	1	1	0	1	0
---	---	---	---	---	---	---	---

x = x & 0x08; ← mask

x

?	?	1	1	1	0	1	0
0	0	0	0	1	0	0	0

 ← mask &

x

0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

```
if(x == 0x08) {  
    ...  
}  
if(x != 0) {  
    ...  
}
```

One Bit Mask patterns

- Bit 0 — 0b00000001 — 0x01
- Bit 1 — 0b00000010 — 0x02
- Bit 2 — 0b00000100 — 0x04
- Bit 3 — 0b00001000 — 0x08
- Bit 4 — 0b00010000 — 0x10
- Bit 5 — 0b00100000 — 0x20
- Bit 6 — 0b01000000 — 0x40
- Bit 7 — 0b10000000 — 0x80

What are these masks in hex?

0	0	0	0	1	0	0	1
---	---	---	---	---	---	---	---

0	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---

0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---

1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

1	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

1	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

0	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---

1	0	1	0	1	0	1	1
---	---	---	---	---	---	---	---