Software Embarcado

02 - GPIO

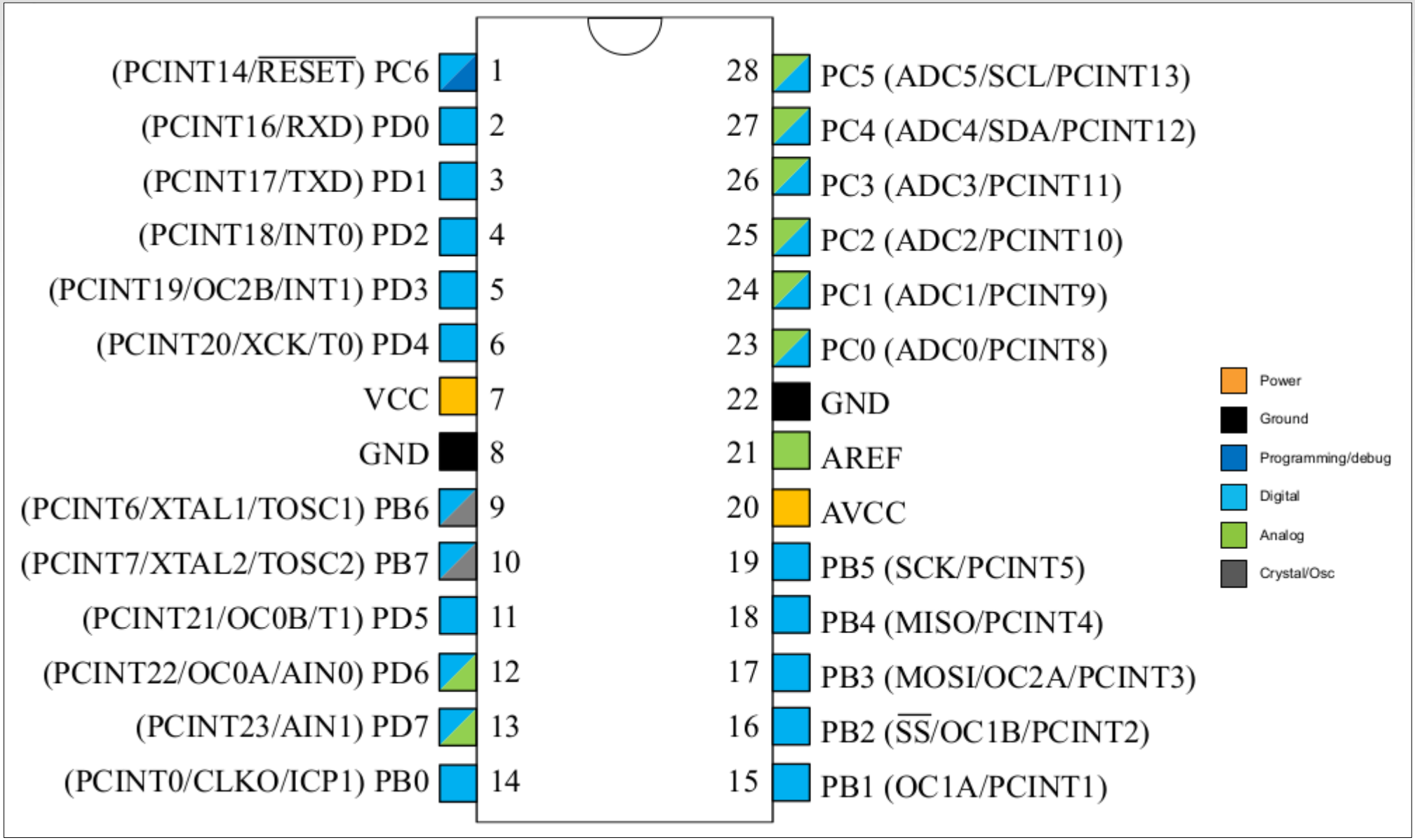
Francisco Sant'Anna
Sala 6020-B
francisco@ime.uerj.br

http://github.com/fsantanna-uerj/SE

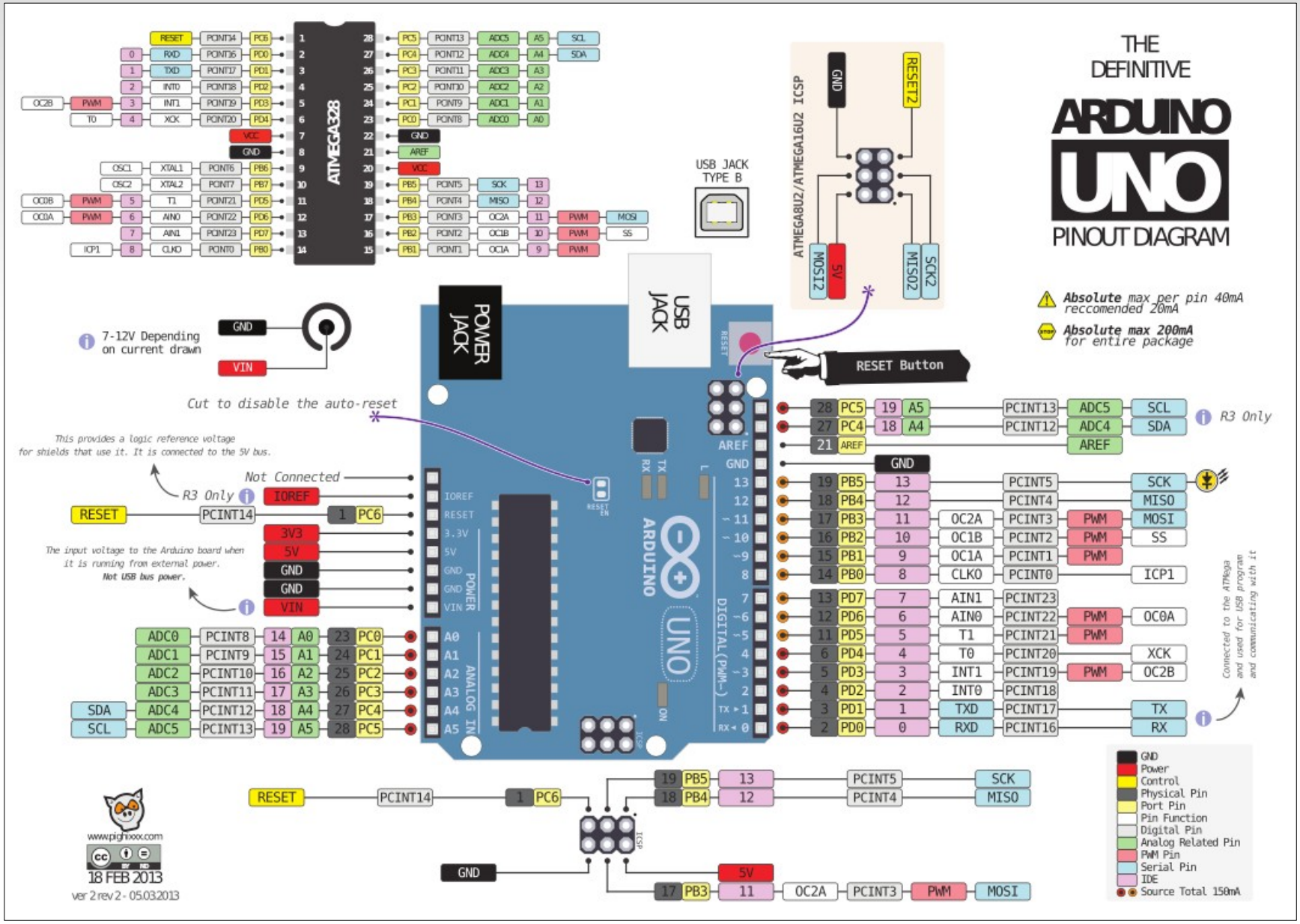
GPIO
General-Purpose Input/Output

- Entrada digital
- Saída digital
- Saída analógica PWM (hw ou sw)

Pinagem do Atmega328p



ATmega328p / Arduino UNO



I/O Multiplexing

- Cada pino está associado a uma porta de GPIO
- Alternativamente, podem ser associados a outras funcionalidades de periféricos

5.2.4. Port C (PC[5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC[5:0] output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

5.2.5. PC6/$\overline{\text{RESET}}$

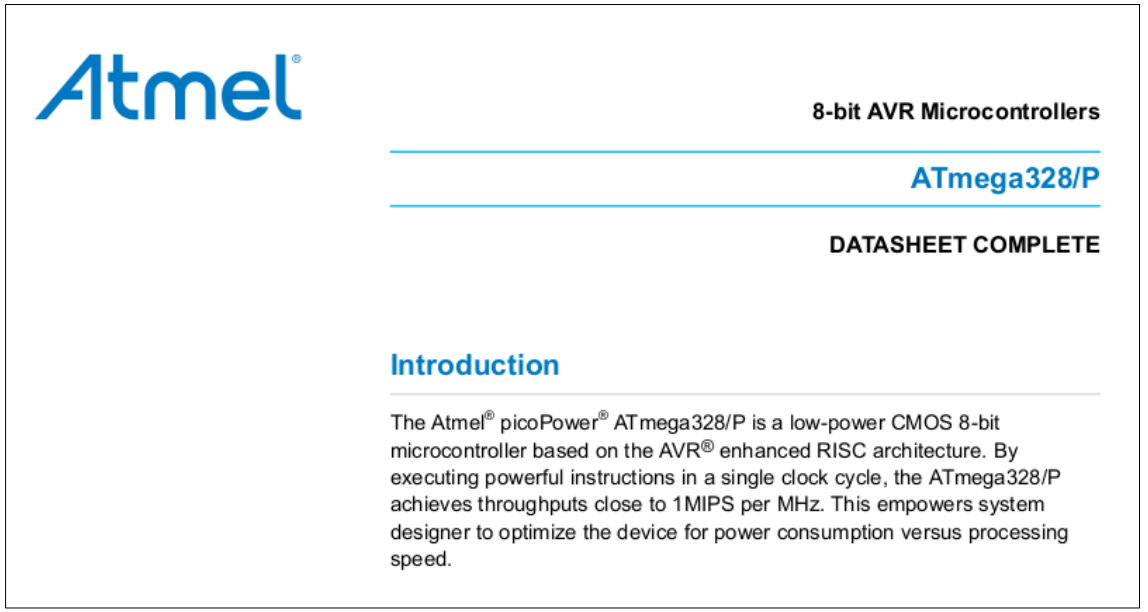
If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in the *Alternate Functions of Port C* section.

Datasheet ATmega328p

- `avr-atmega328p.pdf` (442 págs)
  - Pág 14: pinagem
  - Pág 18: porta C
  - Pág 97: portas



Atmel

8-bit AVR Microcontrollers

ATmega328/P

DATASHEET COMPLETE

Introduction

The Atmel® picoPower® ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

Portas B,C,D

- Cada porta está associada a três registradores de 8bits (`DDRx`, `PORTx`, `PINx`)
- Cada pino individual está associado a um bit em cada registrador:
  - `DDRxn: [w]` direção do pino `n` na porta `x`
  - `PORTxn:[w]` novo valor ou configuração *pull up* do pino `n` na porta `x`
  - `PINxn: [r]` valor atual do pino `n` na porta `x`

Portas B,C,D

Table 18-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Bit	7	6	5	4	3	2	1	0
	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Hello World: output

- Piscar o LED a cada 1 segundo
- `github.com/fsantanna-uerj/SE/tree/master/Code/01-blink/`

```
#define LED_PIN 13

void setup () {
    pinMode(LED_PIN, OUTPUT);      // Enable pin 13 for digital output
}

void loop () {
    digitalWrite(LED_PIN, HIGH);   // Turn on the LED
    delay(1000);                   // Wait one second (1000 milliseconds)
    digitalWrite(LED_PIN, LOW);    // Turn off the LED
    delay(1000);                   // Wait one second
}
```

Usando portas GPIO...

- Piscar o LED a cada 1 segundo
- `github.com/fsantanna-uerj/SE/tree/master/Code/01-blink/`

```
void setup() {
  DDRB = 0b00100000;
}

void loop() {
  PORTB = 0b00100000;
  delay(1000);
  PORTB = 0b00000000;
  delay(1000);
}
```

```
void setup() {
  DDRB = 1 << 5;
}

void loop() {
  PORTB = 1 << 5;
  delay(1000);
  PORTB = 0;
  delay(1000);
}
```

```
void setup() {
  DDRB |= 1 << 5;
}

void loop() {
  PORTB |= 1 << 5;
  delay(1000);
  PORTB &= ~(1 << 5);
  delay(1000);
}
```

- Ver `pinMode`, `digitalWrite`, `digitalRead`
- Tarefa 2 – acesso direto às portas GPIO
  - Entrada e saída