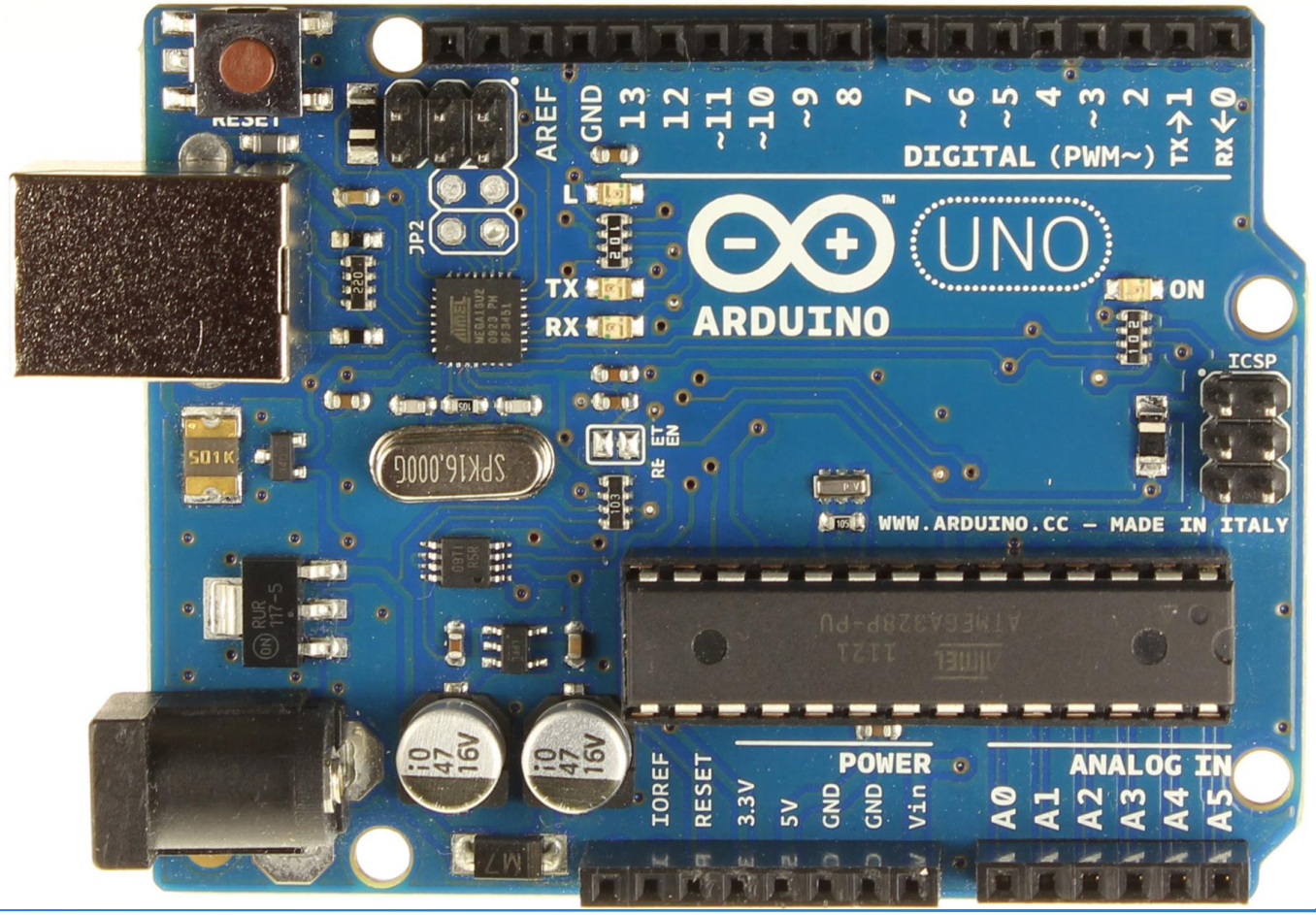
Curso intermedio sobre Arduino: Acceso avanzado a puertos

Elcacharreo.com

Arduino Intermedio: Presente



Arduino Intermedio: Presente

José Antonio Vacas Martínez

blog

javacasm@elcacharreo.com

twitter
linkedin

Arduino Intermedio: Puertos

Arduino UNO

Las E/S en los microcontroladores están agrupadas en puertos

- B (digital pin 8 to 13)
- C (analog input pins)
- D (digital pins 0 to 7)

3 Registros por puerto

DDR	INPUT o OUTPUT
PORT	HIGH o LOW
PIN	estado de las entradas INPUT

Referencia

Arduino Intermedio: Puertos

PORTD maps to Arduino digital pins 0 to 7

DDRD - The Port D Data Direction Register - read/write

PORTD - The Port D Data Register - read/write

PIND - The Port D Input Pins Register - read only

PORTB maps to Arduino digital pins 8 to 13 The two high bits (6 & 7) map to the crystal pins and are not usable

DDRB - The Port B Data Direction Register - read/write

PORTB - The Port B Data Register - read/write

PINB - The Port B Input Pins Register - read only

PORTC maps to Arduino analog pins 0 to 5. Pins 6 & 7 are only accessible on the Arduino Mini

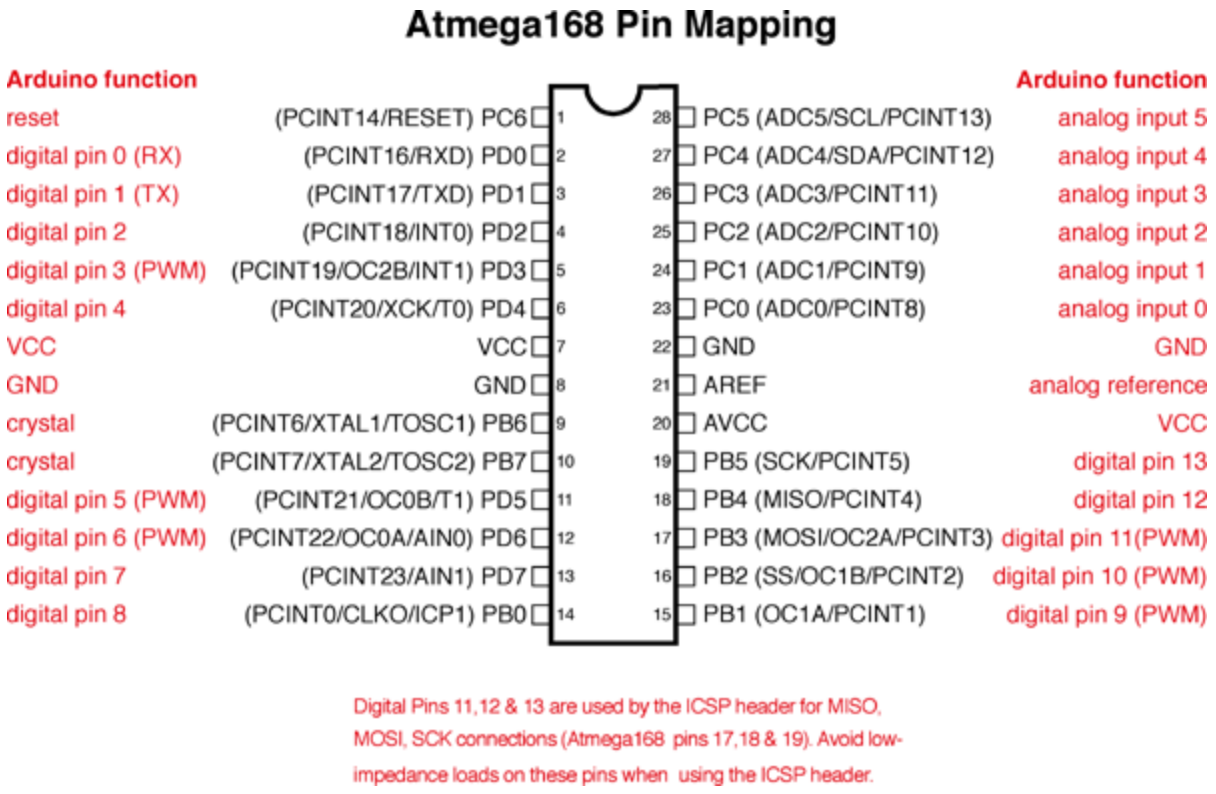
DDRC - The Port C Data Direction Register - read/write

PORTC - The Port C Data Register - read/write

PINC - The Port C Input Pins Register - read only

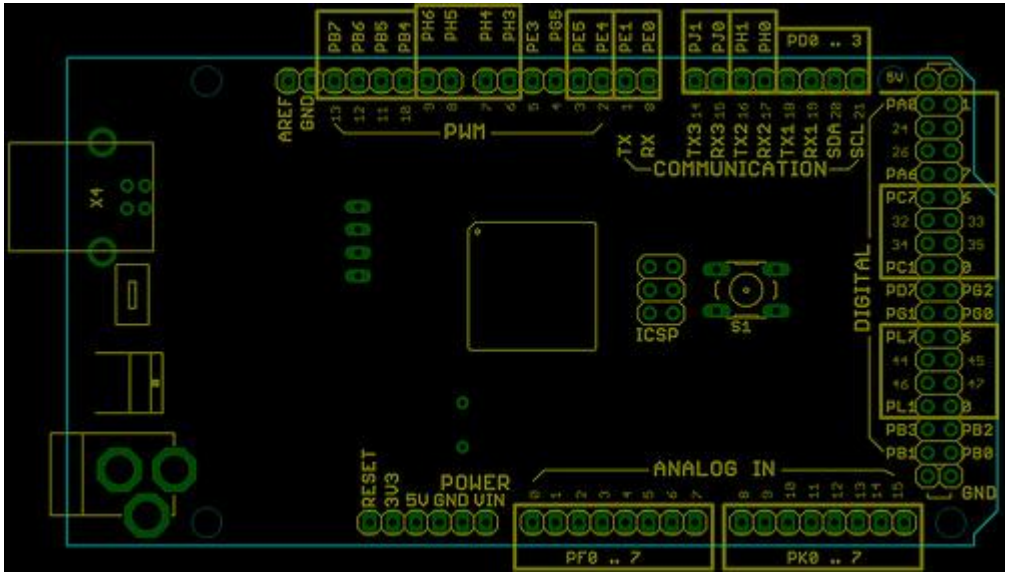
Referencia

Arduino Intermedio: Puertos



http://www.arduteka.com/2013/02/arduino-pinout/

Arduino Intermedio: Puertos



http://www.arduteka.com/2013/02/arduino-pinout/

Arduino Intermedio: Puertos

Un digitalRead() tarda 4134 ns
Un acceso a puerto tarda 83 ns
digitalRead() es **50x** más lento que acceso directo

http://jeelabs.org/2010/01/06/pin-io-performance/

http://hackaday.com/2010/01/06/arduino-io-speed-breakdown/

Acceso a puertos: Ejemplos

PortD mapea Arduino digital pins 0 to 7 (pins 0 & 1 son TX y RX)

DDRD es la dirección del Port D (Arduino digital pins 0-7) controla si PORTD se configura como inputs o outputs so

```
DDRD = B11111110;  // 1 to 7 as outputs, pin 0 as input
```

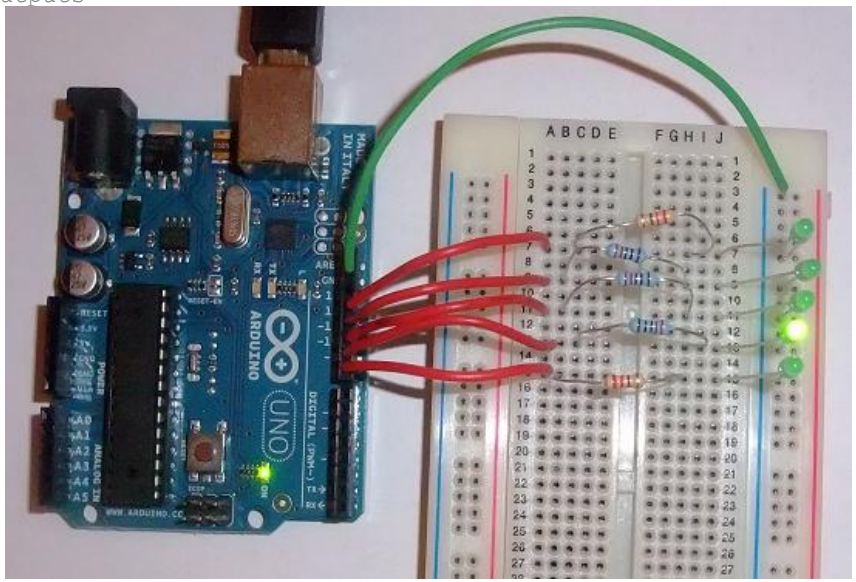
```
DDRD = DDRD | B11111100; asegura que no tocamos 0 y 1
```

```
PORTD = B10101000; // ponemos los pin 7,5,3 HIGH
```

Acceso a puertos: Ejercicios

Kit (cylons)

```
unsigned char upDown=1;    // start off going UP
unsigned char cylon=0;  // determines which LED is on 0 to 4

void setup() {
  // initialize the digital pins as outputs.
  DDRB = B00011111;  // sets Arduino port B pins 0 to 4 as outputs
}

void loop() {
  if(upDown==1){
    cylon++;
    if(cylon>=4) upDown=0;
// Reached max LED, next time we need to go down
  }
  else {
    cylon--;
    if(cylon==0) upDown=1;
// Reached min LED, next time we need to go up
  }
  PORTB = 1 << cylon;
  delay(150);              // wait for a second
}
```



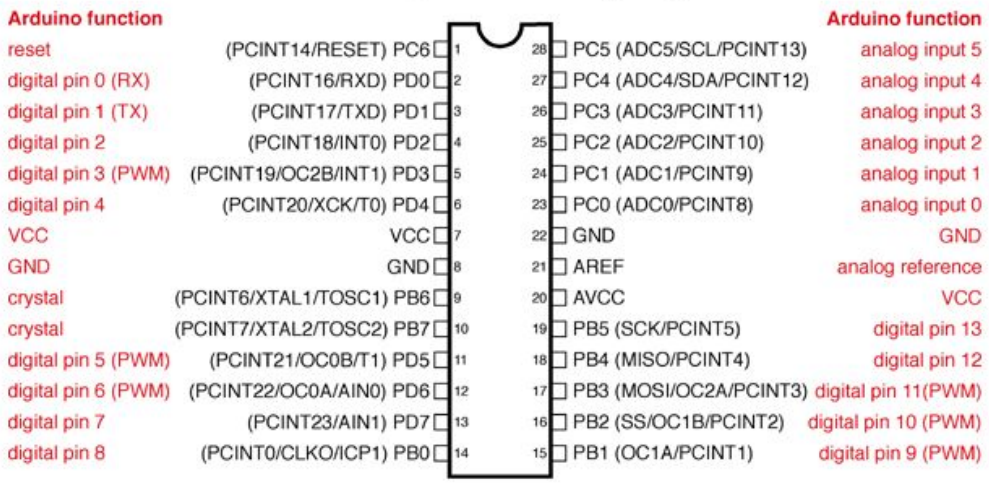
Acceso a puertos: ¿cómo de lento?

```
void setup() {  Serial.begin(9600); }
void loop(){
 int initial = 0;
 int final = 0;
 initial = micros();
 for(int i = 0; i < 500; i++)
 {        digitalWrite(13,HIGH);
          digitalWrite(13,LOW); }
 final = micros();
 Serial.print("Time for digitalWrite(): ");  Serial.print(final-initial);  Serial.println("");
 initial = micros();
 for(int i = 0; i < 500; i++)
 {        PORTB |= _BV(PB5);
          PORTB &= ~_BV(PB5);  }
 final = micros();
 Serial.print("Time for true c command: ");
 Serial.print(final-initial);
 while(1);
}
```

Acceso a puertos: ¿cómo de lento?

```
PORTB |= _BV(PB5);    (high)
PORTB &= ~_BV(PB5); (low)
```

Atmega168 Pin Mapping



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

Acceso a puertos: Ejercicios

```
#include <avr/io.h>                    //This is our usual include
#define F_CPU 16000000UL       //This says to the compiler what frequency is running, 16Mhz
#include <util/delay.h>                //The delay functions/routines

uint8_t readButton(void);              //Declaration of the readButton function

int main(void){
DDRD &= ~(1<<PD2);             // Configure PORTD pin 2 as an input
PORTD |= (1<<PD2);             // Activate pull-ups in PORTD pin 2
DDRB |= (1<<PB5);                      // Configure PORTB pin 5 an output,
                                       // this is the digital 13 in the Arduino that as the built-in led

 while(1){                             //Infinite loop
 if(readButton()==1)                   //Verify the button state
       PORTB ^=(1<<PB5);       //This is the above mentioned XOR that toggles the led
 _delay_ms(250);                       //Delay between consecutive button presses
 }}

uint8_t readButton(void){
 if((PIND & (1<<PD2)) == 0){           //If the button was pressed
 _delay_ms(25); }                      //Debounce the read value
 if((PIND & (1<<PD2)) == 0)            //Verify that the value is the same that what was read
      return 1;                        //If it is still 0 its because we had a button press
 else                                  //If the value is different the press is invalid
      return 0;
 }
```

Internal pull-up

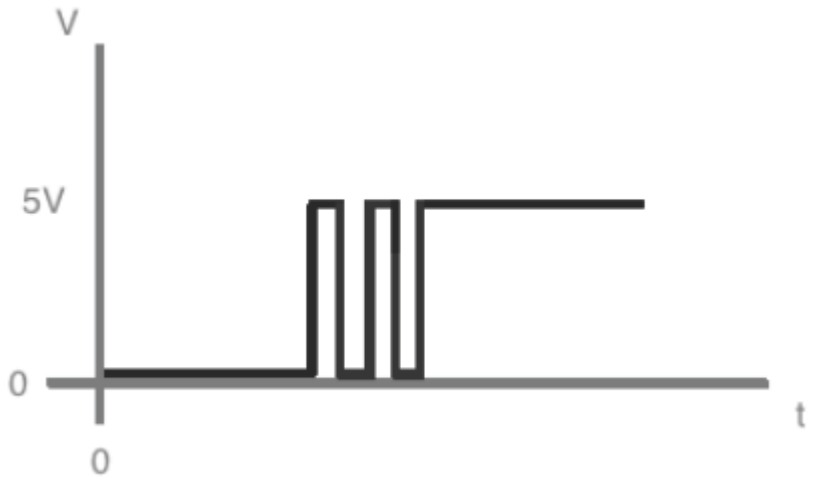
```
void setup(){
  //start serial connection
  Serial.begin(9600);
  //configure pin2 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP);
  pinMode(13, OUTPUT);

}

void loop(){
  //read the pushbutton value into a variable
  int sensorVal = digitalRead(2);
  //print out the value of the pushbutton
  Serial.println(sensorVal);

  // Keep in mind the pullup means the pushbutton's
  // logic is inverted. It goes HIGH when it's open,
  // and LOW when it's pressed. Turn on pin 13 when the
  // button's pressed, and off when it's not:
  if (sensorVal == HIGH) {
    digitalWrite(13, LOW);  }
  else {
    digitalWrite(13, HIGH);  }
}
```

De-bouncing

dealy(10);

condensador

flancos



Arduino Intermedio: Multiplex

Casi siempre faltan entradas o salidas

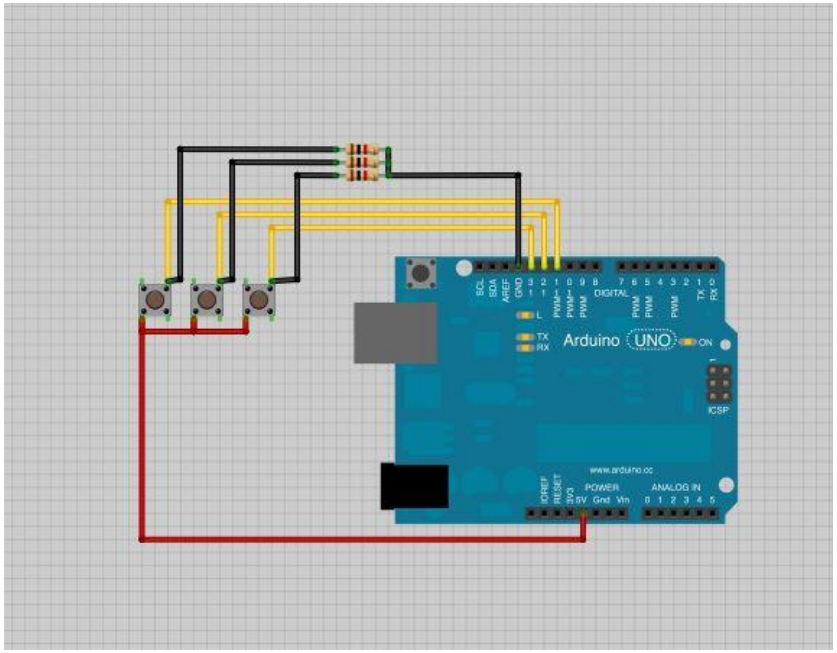
Hay muchas formas de hacer multiplexado de entradas. La mayoría usa un chip al que conectamos nuestras entradas y que incluye una salida (que conectamos a arduino) y unas patillas que usamos para direccionar qué entrada queremos leer.

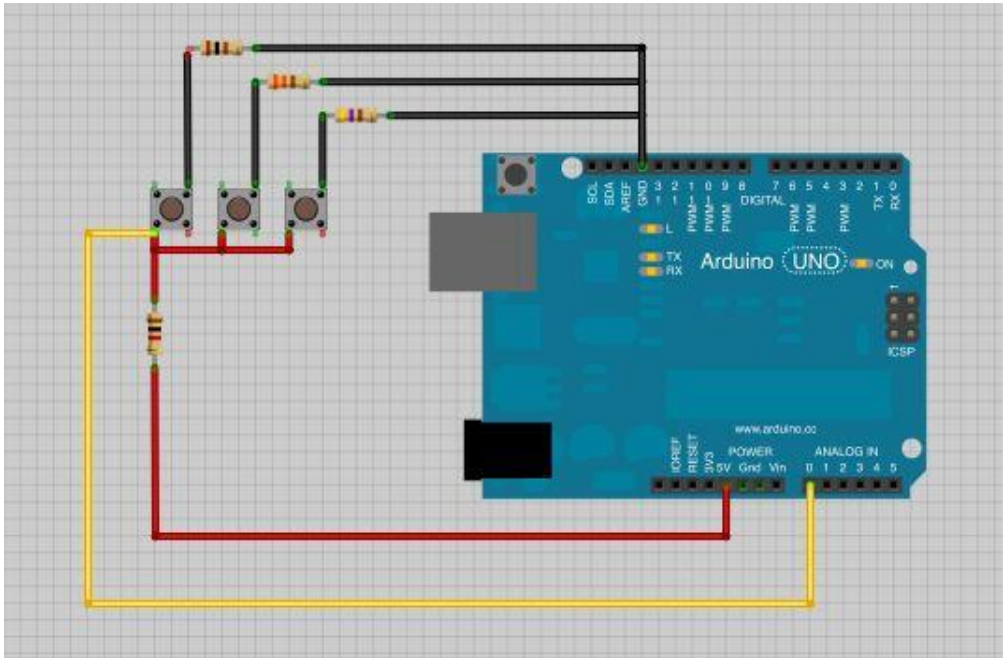
Ejemplos de estos chips son:

* El 4051 que además sirve como multiplexor analógico

* El 151 algo más básico pero que nos sirve como latch, es decir al activar su señal strobe, toma una "foto" del estado de las entradas que nos permite leer luego.

Multiplexar entradas





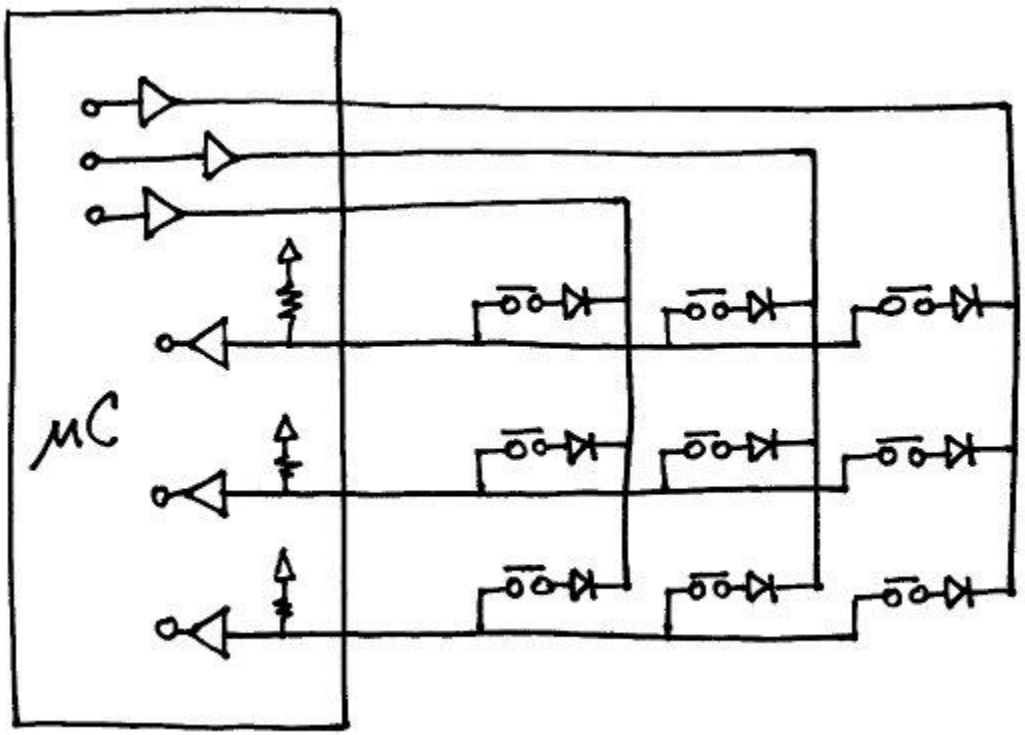
http://www.instructables.com/id/How-to-multiplex-inputs-with-resistors/?ALLSTEPS)

Multiplexar entradas

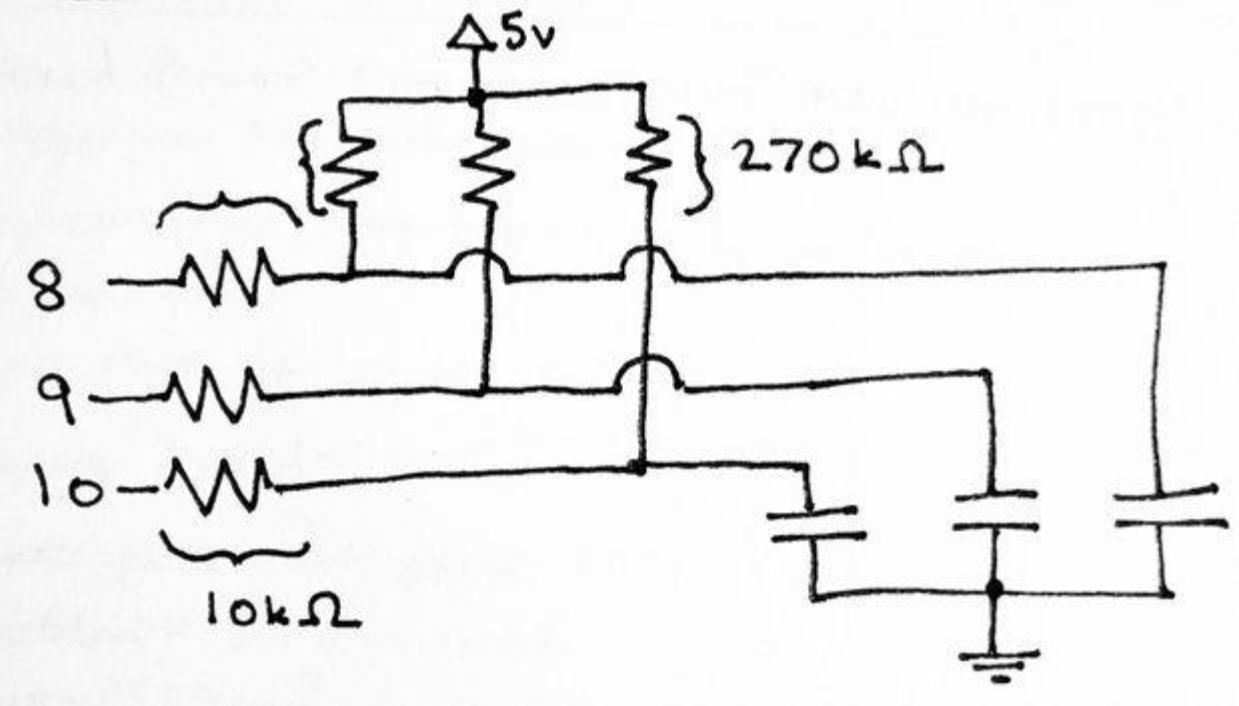
```
int buttonValue1 = 92; //convert 0.45V into analog reading
int buttonValue2 = 254; //convert 1.24V into analog reading
int buttonValue3 = 327; //convert 1.6V into analog reading
int value;
void setup () { }

void loop() {
  value = analogRead(A1);
  switch (value) {
            case 92:
                      //button one was pressed
                      break;
            case 254:
                     //button two was pressed
                      break;
            case 327:
                      //button three was pressed
                      break;
            case 72: //convert 0.35V into analog reading
                      //button one AND button two were pressed
                     break;
            default:
                      //no button pressed
                      break;
      } }
```

http://www.instructables.com/id/How-to-multiplex-inputs-with-resistors/?ALLSTEPS)

Multiplexar entradas: matriz



Entradas táctiles



http://www.instructables.com/id/DIY-3D-Controller/?ALLSTEPS

http://playground.arduino.cc//Main/CapacitiveSensor?from=Main.CapSense

Entradas táctiles: Programa

```
#define resolution 8
#define mains 50 // 60: north america, japan; 50: most other places
#define refresh 2 * 1000000 / mains

void setup() {
  Serial.begin(115200);   // unused pins are fairly insignificant,  but pulled low to reduce unknown variables
  for(int i = 2; i < 14; i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);  }
  for(int i = 8; i < 11; i++ )     pinMode(i, INPUT);
  startTimer();
}

void loop() {
  Serial.print(time(8, B00000001), DEC);
  Serial.print(" ");
  Serial.print(time(9, B00000010), DEC);
  Serial.print(" ");
  Serial.println(time(10, B00000100), DEC);
}
```

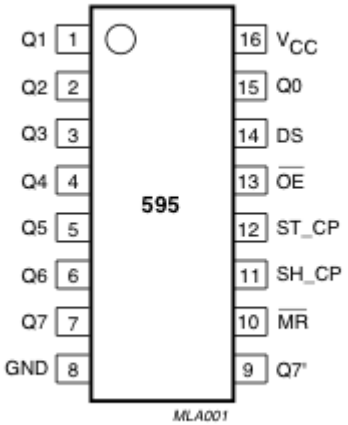
http://www.instructables.com/id/DIY-3D-Controller/?ALLSTEPS

Entradas táctiles: Programa II

```
long time(int pin, byte mask) {
  unsigned long count = 0, total = 0;
  while(checkTimer() < refresh) {
    // pinMode is about 6 times slower than assigning
    // DDRB directly, but that pause is important
    pinMode(pin, OUTPUT);
    PORTB = 0;
    pinMode(pin, INPUT);
    while((PINB & mask) == 0)       count++;
    total++;
  }
  startTimer();
  return (count << resolution) / total;}

extern volatile unsigned long timer0_overflow_count;

void startTimer() {   timer0_overflow_count = 0;      TCNT0 = 0;}

unsigned long checkTimer() {   return ((timer0_overflow_count << 8) + TCNT0) << 2; }
```

http://www.instructables.com/id/DIY-3D-Controller/?ALLSTEPS

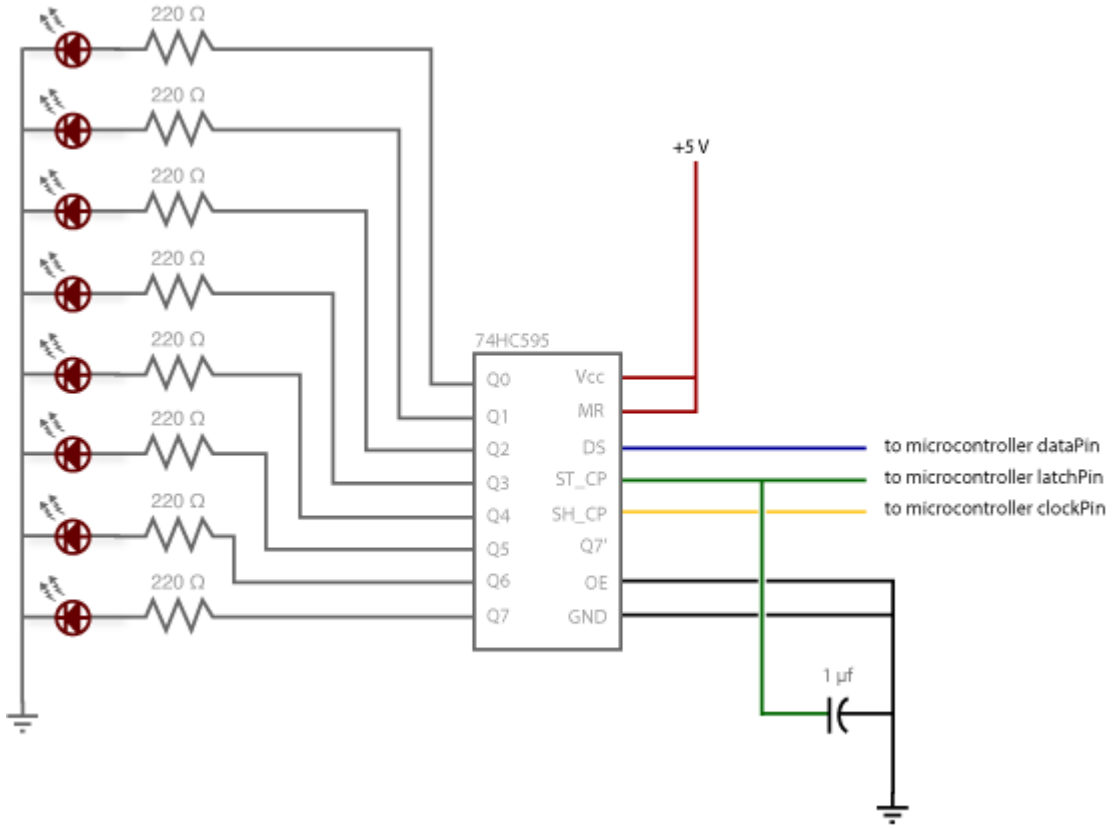
Multiplexar salidas: 595



PINS 1-7, 15	Q0 " Q7	Output Pins
PIN 8	GND	Ground, Vss
PIN 9	Q7"	Serial Out
PIN 10	MR	Master Reclear, active low
PIN 11	SH_CP	Shift register clock pin
PIN 12	ST_CP	Storage register clock pin (latch pin)
PIN 13	OE	Output enable, active low
PIN 14	DS	Serial data input
PIN 16	Vcc	Positive supply voltage

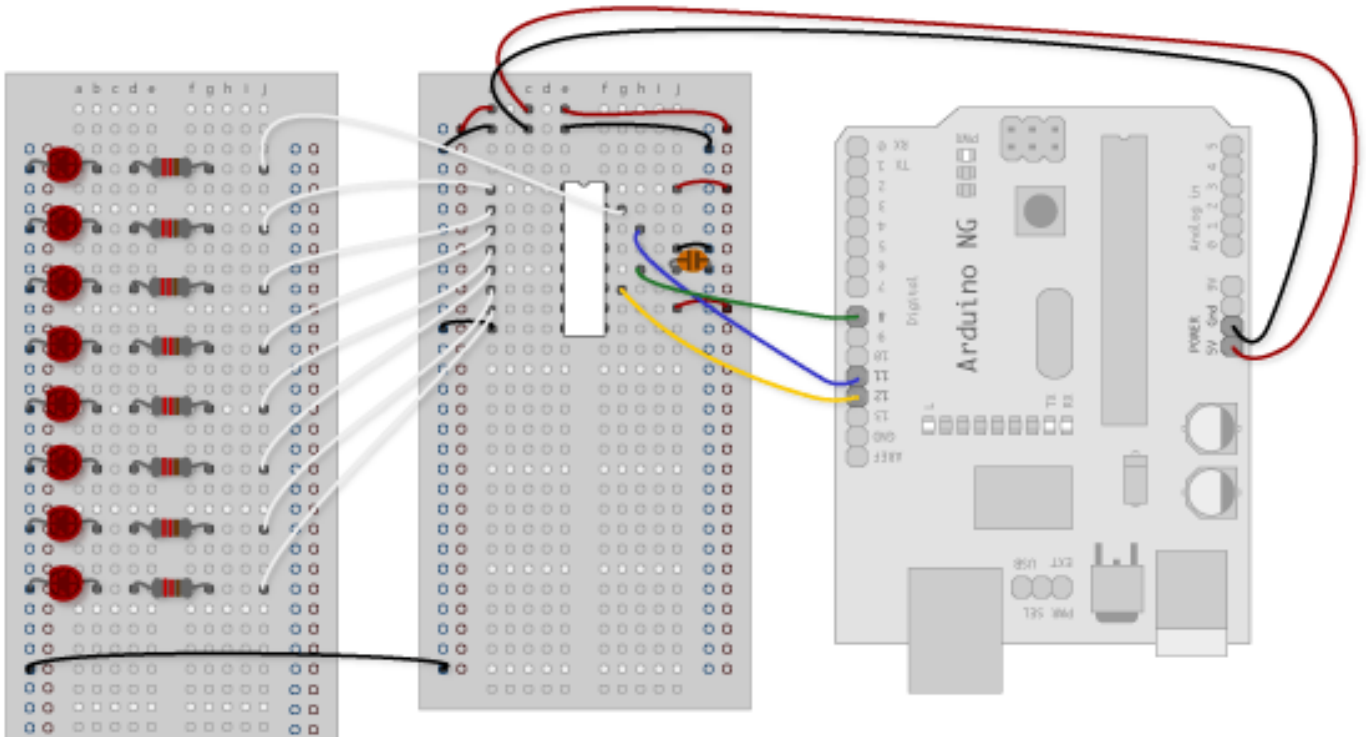
Ejemplo

Multiplexar salidas:595



Ejemplo

Multiplexar salidas:595

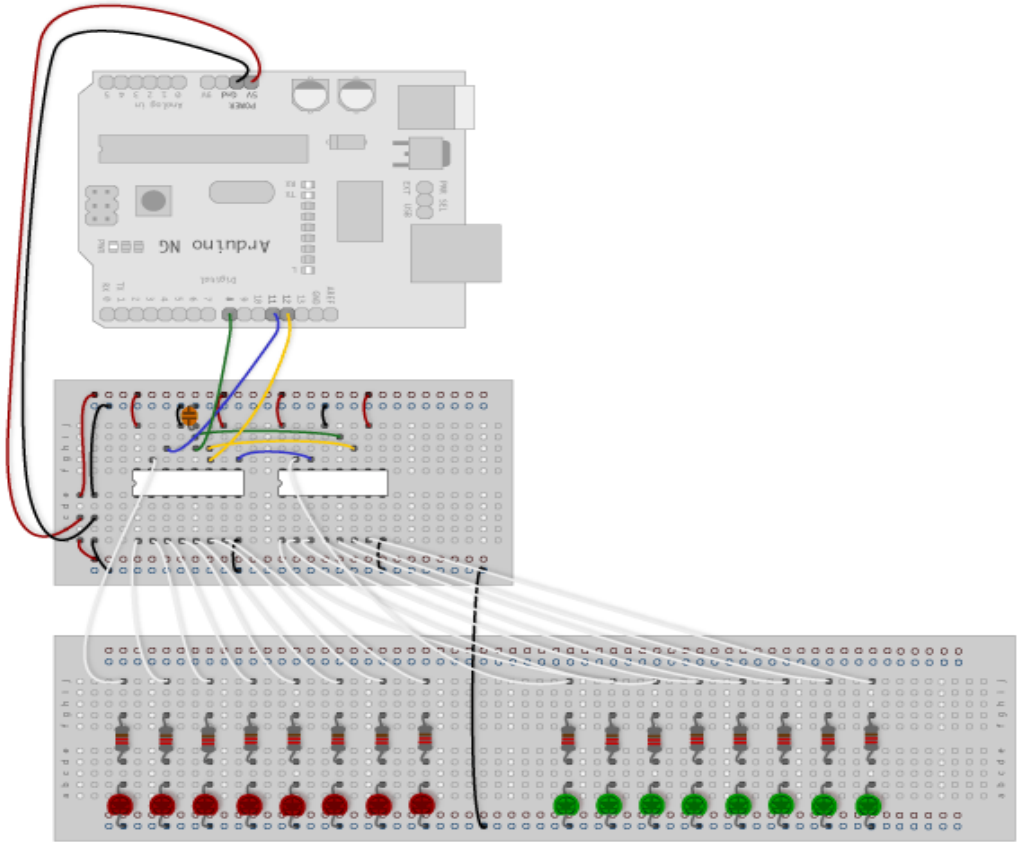


Ejemplo

Multiplexar salidas:595

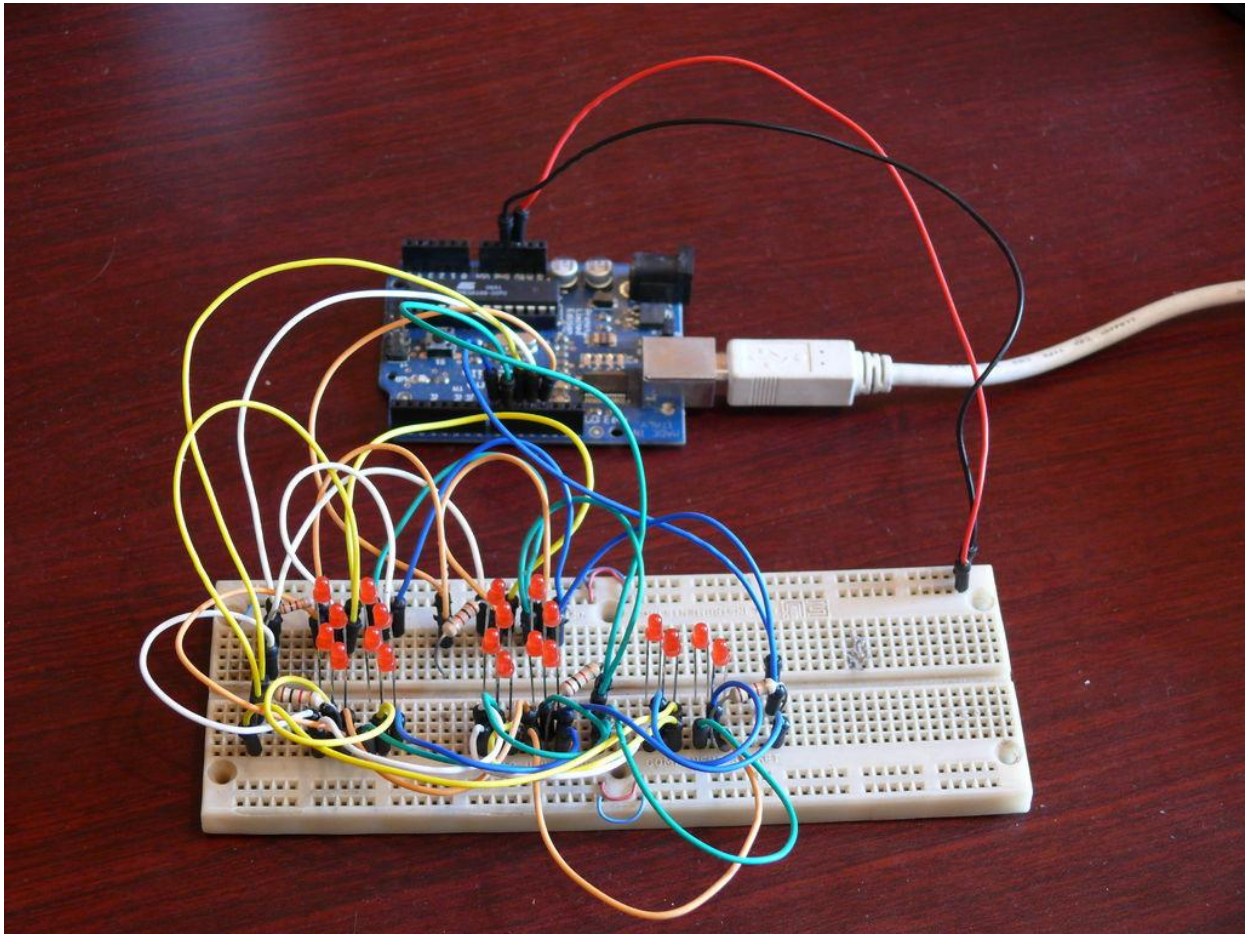
```
int latchPin = 8; //Pin connected to ST_CP of 74HC595
int clockPin = 12; //Pin connected to SH_CP of 74HC595
int dataPin = 11;  ////Pin connected to DS of 74HC595
void setup() {
  //set pins to output so you can control the shift register
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}
void loop() {
  // count from 0 to 255 and display the number
  // on the LEDs
  for (int numberToDisplay = 0; numberToDisplay < 256; numberToDisplay++) {
    // take the latchPin low so
    // the LEDs don't change while you're sending in bits:
    digitalWrite(latchPin, LOW);
    // shift out the bits:
    shiftOut(dataPin, clockPin, MSBFIRST, numberToDisplay);

    //take the latch pin high so the LEDs will light up:
    digitalWrite(latchPin, HIGH);
    // pause before next value:
    delay(500);
  }
}
```

Multiplexar salidas: 2x 595



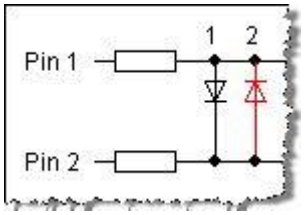
Multiplexar salidas: Charlieplexing

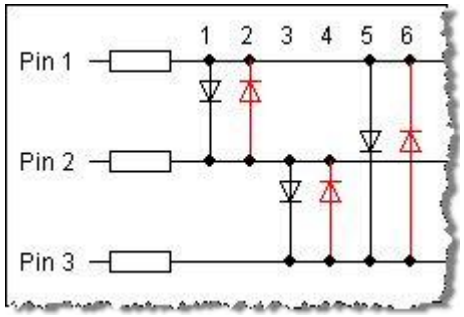
20 leds con 5 salidas
n*(n-1) LEDs en n pins

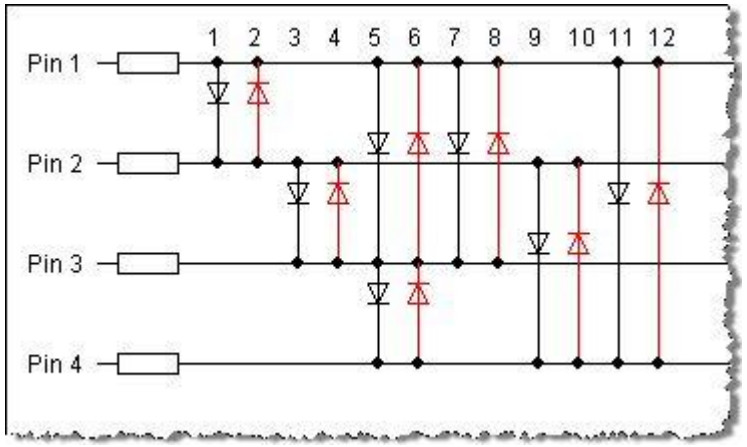


Multiplexar salidas: Charlieplexing

20 leds con 5 salidas (usamos 3 estados....)
n*(n-1) LEDs en n pins







http://www.instructables.com/id/Controlling-20-Leds-from-5-Arduino-pins-using-Cha/?ALLSTEPS

Conclusiones

Gracias por vuestra atención