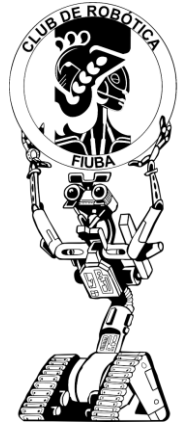


Competencia de robótica

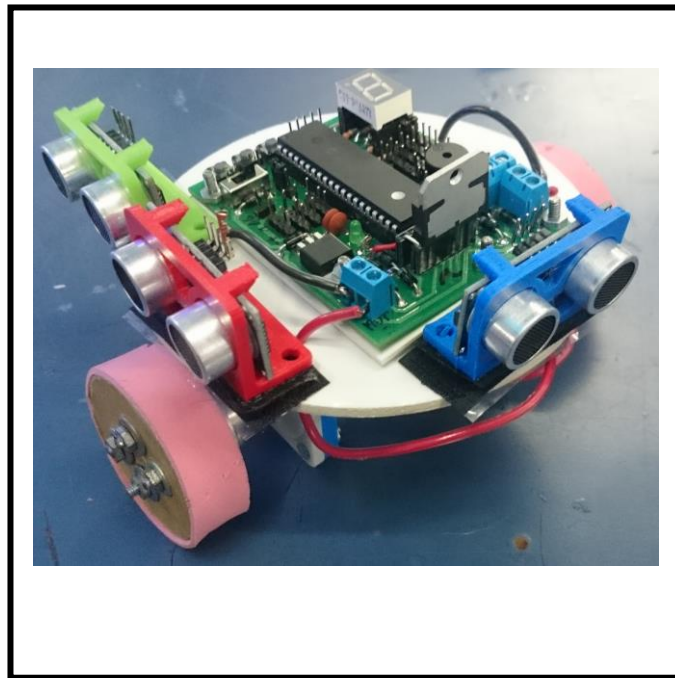
JOHNNY 5

~2015~



Categoría: **Laberinto**

Nombre robot: **Minotauro**



Institución: **Escuela ORT, sede Almagro**
(Mail de contacto: mfoulier@ort.edu.ar)

Participantes: (Nombre completo y mail)

- **Bruno Glicer**
- **Valentina Varela**
- **Melanie Maciel**

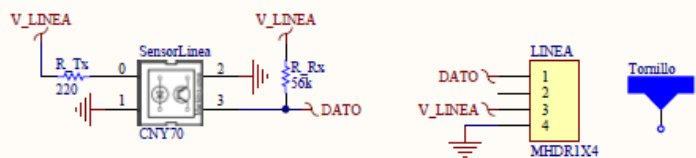
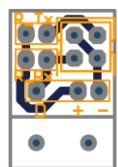
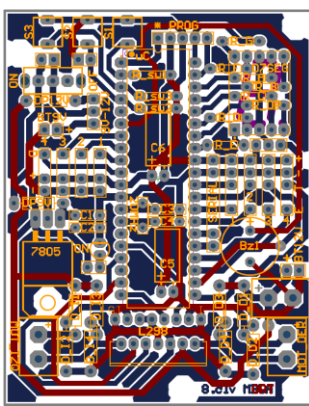
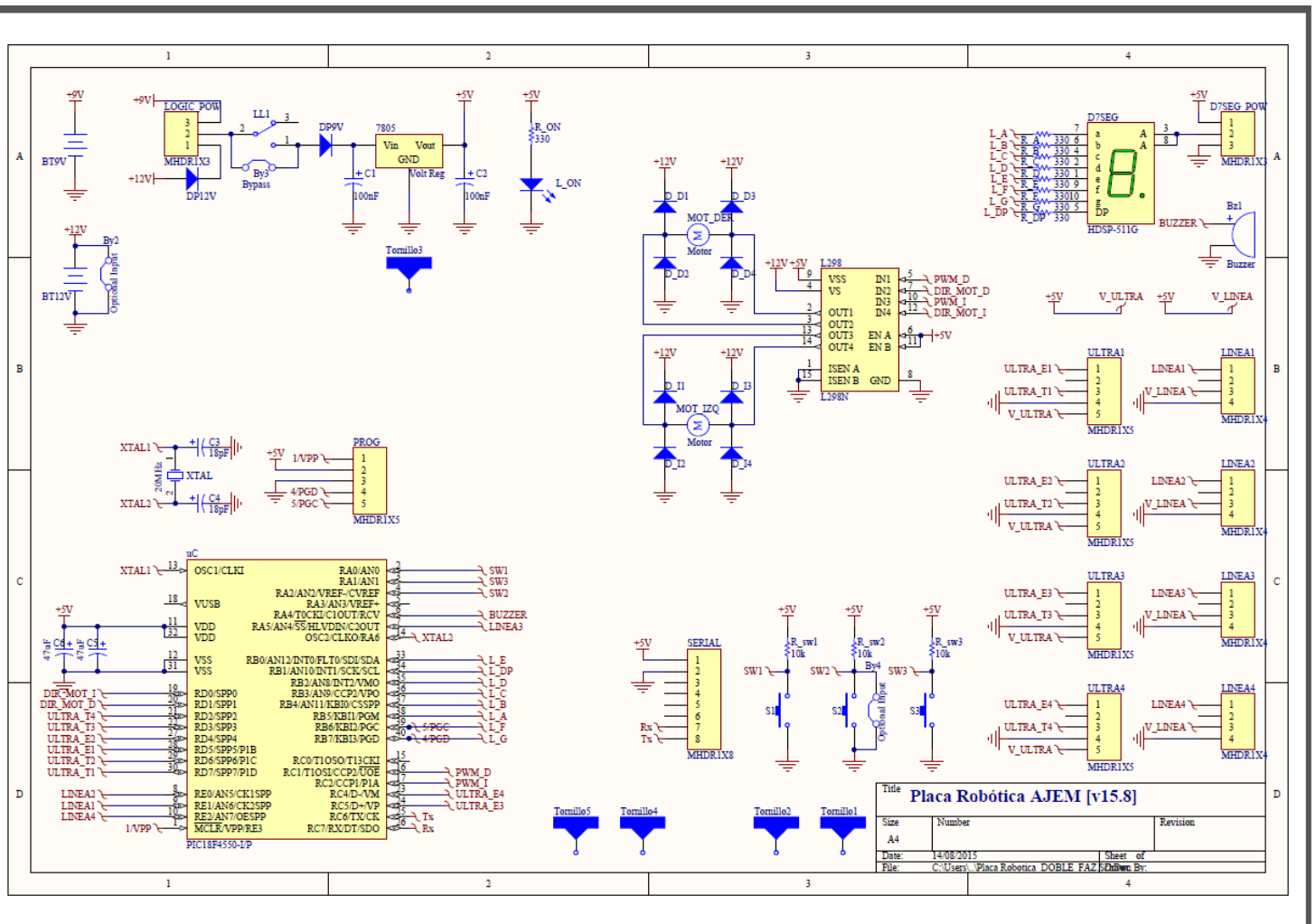
Mecánica:

Diámetro: 12 cm
 Altura: 11cm
 Peso: 500g

Electrónica:

Microcontrolador: PIC18F4550
 Driver motores: L298
 Alimentación: 9V-12V
 Puertos hasta 4 sensores ultrasónicos más 4 sensores infrarrojos
 3 botones de propósito general
 Motores: 2x MR4

Circuitos esquemáticos:



Código fuente:

```
#include "Main.h"
char flag;
void main(void)
{
    unsigned int i;
    int vel_der, vel_izq;
    char Estado=RESET;
    Inicializar();
    test();
    while(1)
    {switch(Estado)
        {
            case RESET:
                vel_izq = -500;
                vel_der = 500;
                StartTimer(TIMER_INICIO, 500);
                Estado = MENU;
                break;

            case MENU:
                testeo_sensores();
                if(TimerIsReady(TIMER_INICIO))
                {
                    setGPIO(LED_DP, IPIN_LED_DP);
                    StartTimer(TIMER_INICIO, 500);
                }
                if(getGPIO(SW1) == PULSADO || getGPIO(SW3) == PULSADO)
                    beep_NO();

                if(getGPIO(SW2) == PULSADO)
                {
                    StopTimer(TIMER_INICIO);
                    i = 5;
                    beep_SI();
                    StartTimer(TIMER_CUENTA_REGRESIVA, 0);

                    Estado = CUENTA_REGRESIVA;
                }
                break;

            case CUENTA_REGRESIVA:
                if(TimerIsReady(TIMER_CUENTA_REGRESIVA))
                {
                    StartTimer(TIMER_CUENTA_REGRESIVA, 980);
                    beep_ALIVE();
                    displayNum(i);
                    if(!i)
                    {
                        StartTimer(TIMER_PWM, 10);
                        Estado = ANDANDO;
                    }
                    i--;
                }
                break;

            case ANDANDO:
                if(TimerIsReady(TIMER_PWM))
                {
                    StartTimer(TIMER_PWM, 10);

                    vel_izq += 10;
                    vel_der += 10;
                    if(vel_izq >= 1000)
                        vel_izq = -1000;
                    if(vel_der >= 1000)
                        vel_der = -1000;

                    set_motores(vel_izq, vel_der);
                }

                if(getGPIO(SW3) == PULSADO)
                {
                    vel_izq += 25;
                    vel_der += 25;
                    set_motores(vel_izq, vel_der);
                    beep_SI();
                    firulete(30, 1);
                }

                if(getGPIO(SW2) == PULSADO)
                {
                    beep_SI();

                    Estado = RESET;
                }

                if(getGPIO(SW1) == PULSADO)
                {
                    vel_izq -= 25;
                    vel_der -= 25;
                    beep_SI();
                    set_motores(vel_izq, vel_der);
                    firulete(30, 1);
                }
                break;

            default:
                Estado = RESET;
                break;
        }
    }
}

void testeo_sensores(void)
{
    /*
    aux = 0x00;
    for(i=0; i<CANT_SENS_PROX; i++)
        if( IRMIsActive(i) )
            aux = aux || 0x01<<i;

    displayByte(aux);
    */
}
```

```

*/
    displayByte(0x01 << 6);
}

void test(void)
{
    unsigned int i;

    i = 0;
    flag = 1;
    setGPIO(PROX3, OFF);
    setGPIO(LED_DP, OFF);

    while(1)
    {
        UltraSonic();

        if(USisDetecting(3))
            setGPIO(LED_DP, ON);
        else
            setGPIO(LED_DP, OFF);

        if(TimerIsReady(0))
        {
            StartTimer(1, 30);

            setGPIO(PROX3, ON);
            setGPIO(LED_DP, ON);
        }

        if(TimerIsReady(1))
        {
            setGPIO(PROX3, OFF);
            setGPIO(LED_DP, OFF);
            flag = 1;

            i++;
            i %= 10;
            displayNum(i);
            //printf("Sensor %u: %04d\n\r", 4, USisDetecting(3));
            //i %= CANT_SENS_PROX;
        }
    }
}

void UltraSonic(void)
{
    static char state = ULTRA_RESET;
    unsigned char i;

    switch(state)
    {
        case ULTRA_RESET:
            TMR0ON = 0; //Stops Timer0
            setTimer0Value(0x0000);

            setGPIO(PROX4, OFF);
            stSensProx[PROX4].last_input = OFF;
            stSensProx[PROX4].detected = 0;
            printf("ULTRA_RESET: Sensor %d: Trigger OFF; Detected = %04d\n\r", PROX4+1, stSensProx[PROX4].detected);

            state = ULTRA_TRIGGER_ON;
            TMR0ON = 1; //Starts Timer0
            break;

        case ULTRA_TRIGGER_ON:
            //printf("ULTRA_TRIGGER_ON: Sensor %d: Trigger ON\n\r", PROX4+1);
            setGPIO(PROX4, ON);
            for(i=0; i<15; i++) //Delay bloqueante de 10 uS
                setGPIO(PROX4, OFF);
            //printf("ULTRA_TRIGGER_ON: Sensor %d: Trigger OFF\n\r", PROX4+1);

            setTimer0Value(0x0000);
            state = ULTRA_ECHO_CHECK;
            if(flag)
            {
                StartTimer(0, 10);
                flag = 0;
            }
            break;

        case ULTRA_ECHO_CHECK:
            if(getTimerValue(0) < T_ULTRA_MAX_DELAY) //Si todavia estoy dentro del tiempo de medicion
            {
                stSensProx[PROX4].input = getGPIO(PROX4);

                //Si hay flanco positivo
                if(stSensProx[PROX4].input == ON && stSensProx[PROX4].last_input == OFF)
                {
                    stSensProx[PROX4].count = getTimerValue(0); //Leo el t_inicial
                    stSensProx[PROX4].last_input = stSensProx[PROX4].input;
                    //printf("ULTRA_ECHO_CHECK: Sensor%u: Flanco Ascendente! t_inicial=%05u\n\r", PROX4+1, stSensProx[PROX4].count);
                    printf("FS:%05u\t", stSensProx[PROX4].count);
                }

                //Si hay flanco negativo
                if(stSensProx[PROX4].input == OFF && stSensProx[PROX4].last_input == ON)
                {
                    stSensProx[PROX4].count = getTimerValue(0) - stSensProx[PROX4].count; //Hago (t_final - t_inicial)
                    stSensProx[PROX4].last_input = stSensProx[PROX4].input;
                    //printf("ULTRA_ECHO_CHECK: Sensor%u: Flanco Descendente! t_medido=%05u\n\r", PROX4+1, stSensProx[PROX4].count);
                    printf("FB:%05u\n\r", stSensProx[PROX4].count);
                }
            }
            else
            {
                //printf("ULTRA_ECHO_CHECK: Timeout!!\n\r");
                stSensProx[PROX4].detected = stSensProx[PROX4].count; //Hago la conversion a centimetros
                //printf("ULTRA_ECHO_CHECK: Sensor%u: detected = %05u\n\r", PROX4+1, stSensProx[PROX4].detected);
                printf(">>TO:%05u\n\r", stSensProx[PROX4].detected);

                //printf("FIN DE CICLO\n\r");

                state = ULTRA_TRIGGER_ON;
            }
            break;
    }
}
}

```