RoboSapien Powered by MSP430

Pedro Dinis Gaspar, António Espírito Santo,

Humberto Santos, Bruno Ribeiro

Engineer/Assistant Professor

Tiago Godinho, Filipe Martins, Luís Costa

Last year Electromechanical Engineering Students

Agenda

- **MSP430@UBI-DEM:**
  - Electromechanical engineering classes and,
  - Last year projects

- Electromechanical Engineering classes - Robotics:
  - **RoboSapien powered by MSP430**

- Last year projects:
  - **Sounds Recording and Reproduction System based in MSP430 for the RoboSapien**

- Summary

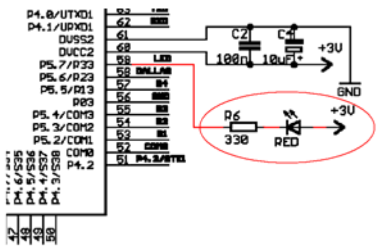
MSP430@UBI-DEM

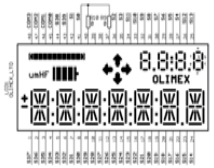
- The MSP430 microcontroller family is used in the Electromechanical Engineering Department of the University of Beira Interior (DEM-UBI) – Portugal in:
  - Electromechanical engineering classes and,
  - last year projects

- Due to:
  - its cost;
  - plenty of hardware starter kits and software integrated development environments either free or with a reduced cost;
  - Amount of peripherals integrated in the μC;
  - rapid learning curve;
  - low power features;
  - high pinout counting;
  - C and C++ easy programming.
  - Among others…

MSP430@UBI-DEM

- Electromechanical Engineering classes
- **Introduction to Microprocessors:**
  - Description of the various systems embedded in a microcontroller;
  - Architecture characteristics;
  - Memory Hierarchy;
  - Internal modules;
  - Input/Output ports;
  - Peripherals;
  - Programming language Assembler / C;
  - Arithmetic;
  - Communications networks.

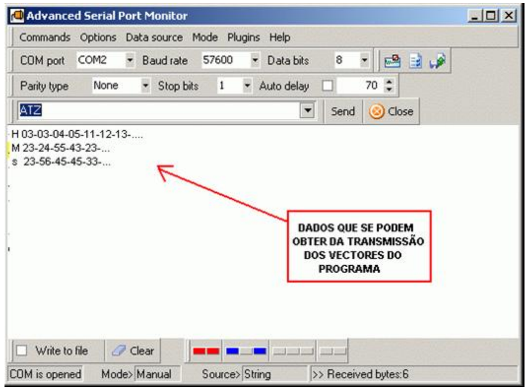




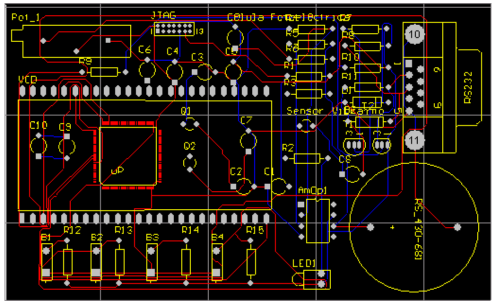
- **All these topics use MSP430 as example**
- **Laboratorial classes demonstrating its capabilities**

MSP430@UBI-DEM

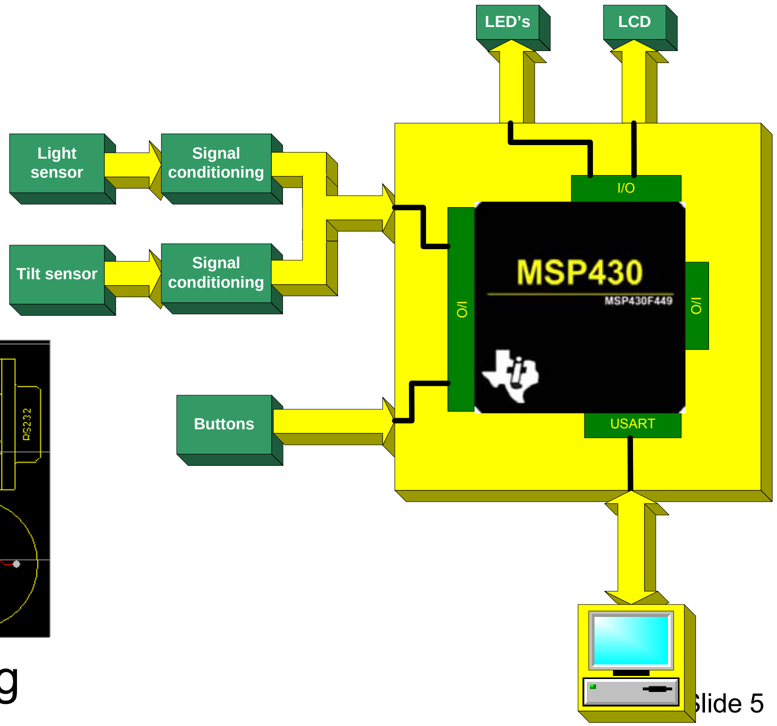
- Electromechanical Engineering classes
- **Instrumentation and Data Acquisition (Lab examples):**
  - MSP430 Monitoring System - Develop a low cost system for event record.



PC data transmission

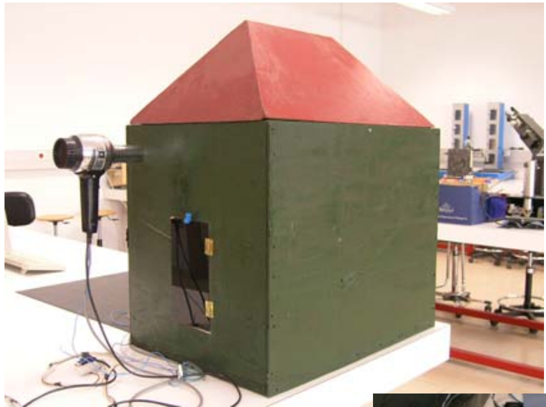


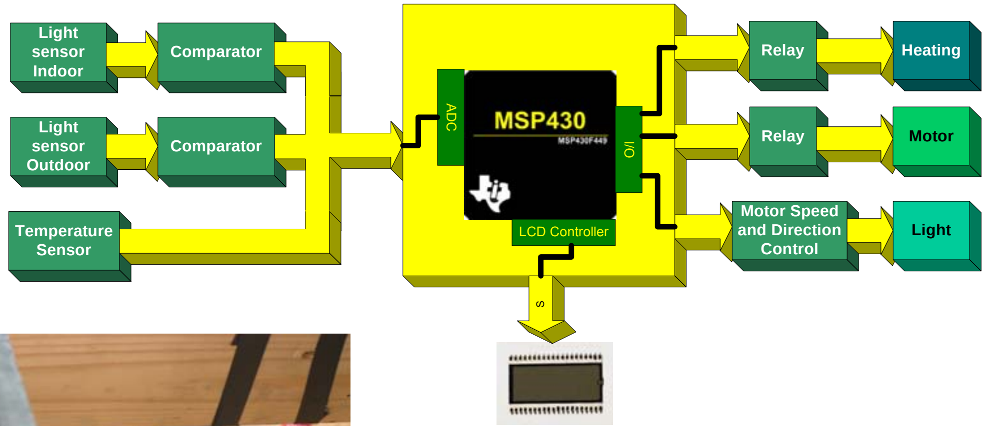
PCB board routing

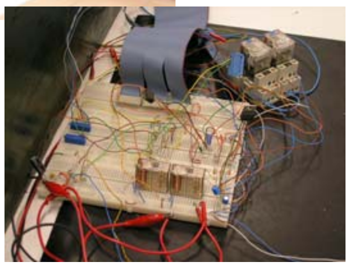


MSP430@UBI-DEM

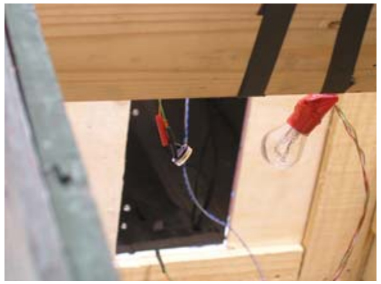
- **Electromechanical Engineering classes**
- **Automation (Lab examples):**
  - MSP430 home heating, illumination and security control.







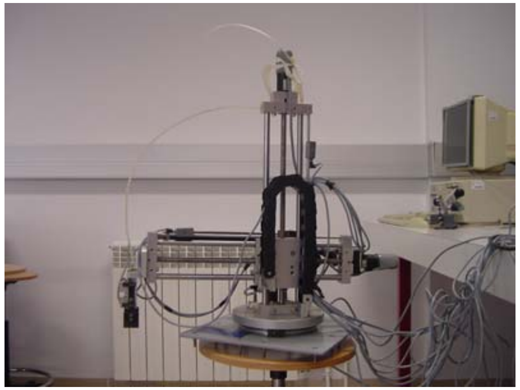
Development board

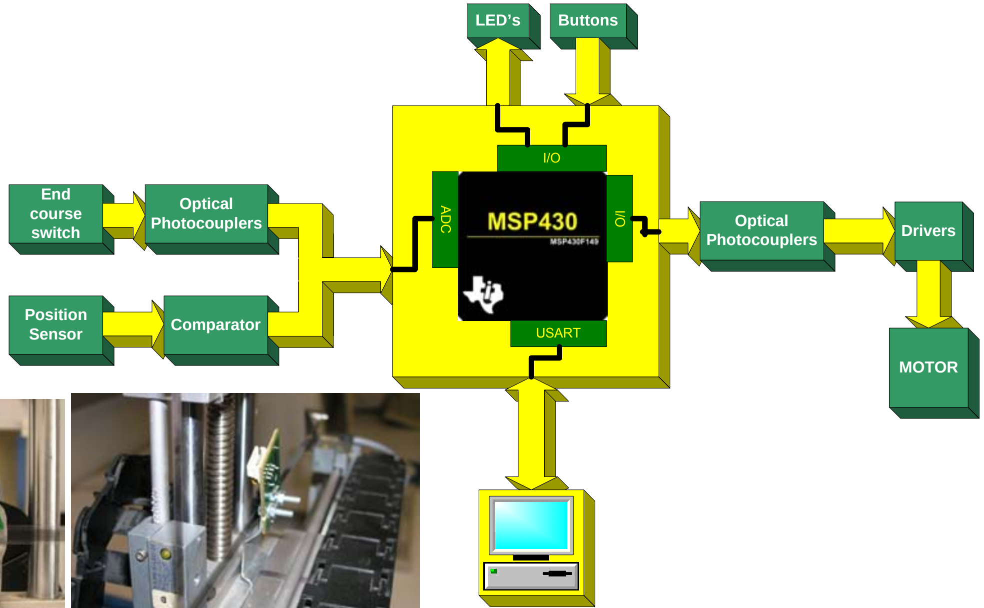


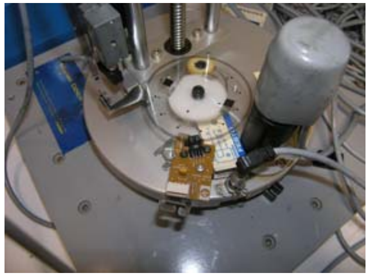
Indoor light sensor

MSP430@UBI-DEM

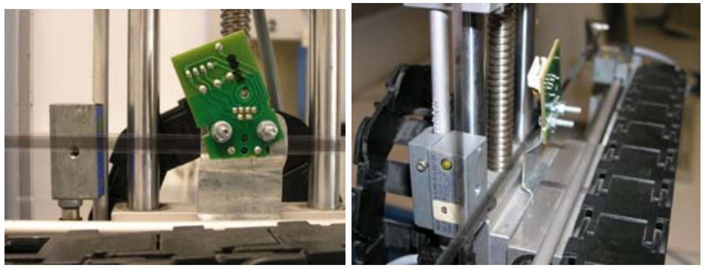
- Electromechanical Engineering classes
- **Robotics (Lab examples):**
  - MSP430 control board and software for the FESTO robot.







Rotational optical sensor



Linear optical sensor for longitudinal movement

MSP430@UBI-DEM

- **Electromechanical Engineering classes - Robotics:**
  - 2006 Project - RoboSapien powered by TI MSP430;
  - Born by the collaboration between DEM-UBI and TI;
  - Substitution of RoboSapien's control/regulation electronics;
  - Replicate the RoboSapien operation;
  - Evidencing the capabilities of TI MSP430;
  - To motivate the students and to develop a technology demo "vehicle".

RoboSapien powered by MSP430

- The tasks involved in the project consisted in the:

(1) analysis of the robot dynamics (evaluation of the robot movements and its characteristics);

(2) analysis of all sensor/actuator devices on the robot and respective signal conditioning and regulation electronics;

(3) development of the hardware (PCB boards and electronics) and software (C code for the controlling program in the MSP430 to commit robot dynamics); and

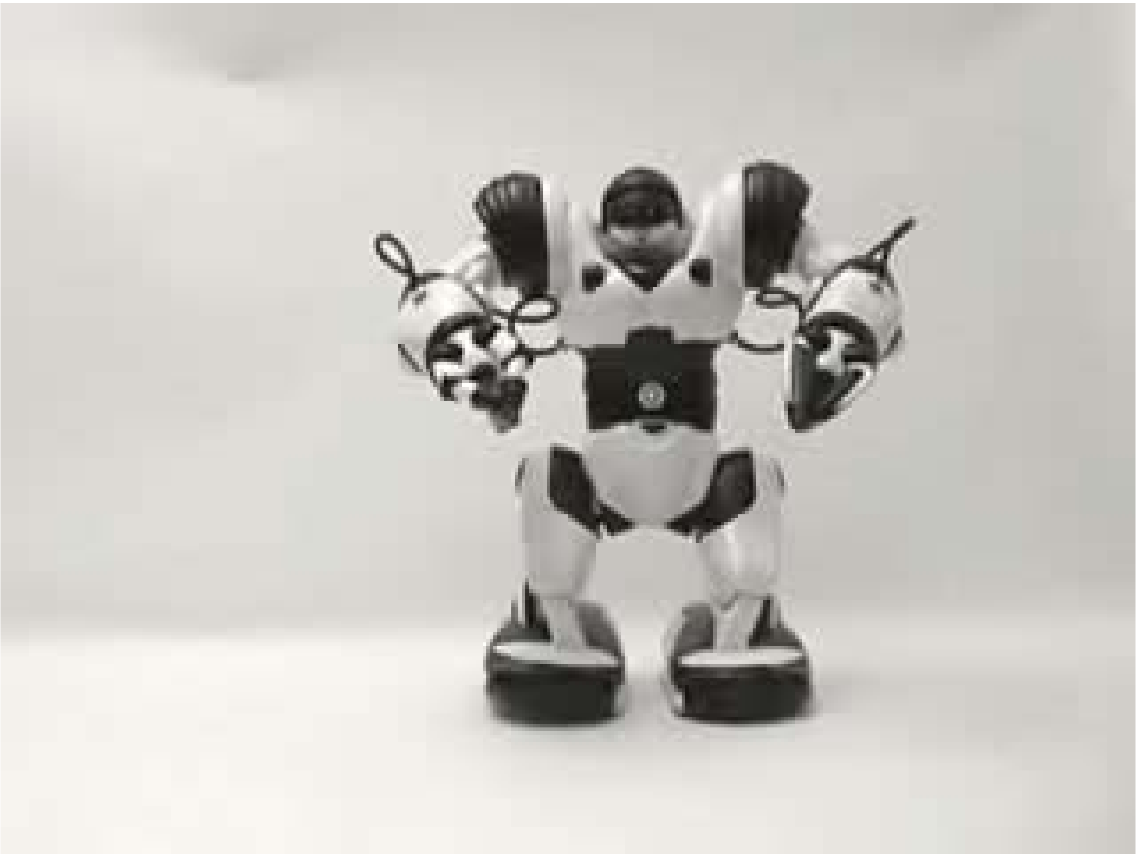
(4) tests and development of new functionalities.

Who is RoboSapien?

- RoboSapien (RS) is a toy-like biomorphic robot.
- Is preprogrammed with moves, and also controlled by an IR remote control;
- It is capable of:
  - a walking motion;
  - grasp objects with either of its hands
  - throw grasped objects with mild force;
- It has a small loudspeaker unit, which can broadcast several different vocalizations.

Who is RoboSapien?

- A brief presentation of the RoboSapien operation.



Who is RoboSapien?

- RoboSapien views.

Frontal view, seven motors move the robot.

Side view.

Who is RoboSapien?

- Dynamic walking gait of RoboSapien.

(1) The trunk motor tilts the upper body to the right. The center of mass shifts over the right foot. The left foot lifts from the ground.

(2) The leg motors move into opposite directions, resulting in a forward motion of the robot. As the upper body swings back, the left foot regains contact with the ground.

(3,4) Symmetrical to (1,2).

Programming of the RoboSapien

- The RS controller, is equipped with a basic level of programmability.

- Users can string together movement commands to form either macros or mini-programs (instructions sets).

- Broadcast of an instruction-set to the RS, save it on-board memory for later execution.

- Sensor-keyed instruction set, performing a specific set of actions on contact with a specific sensor system.

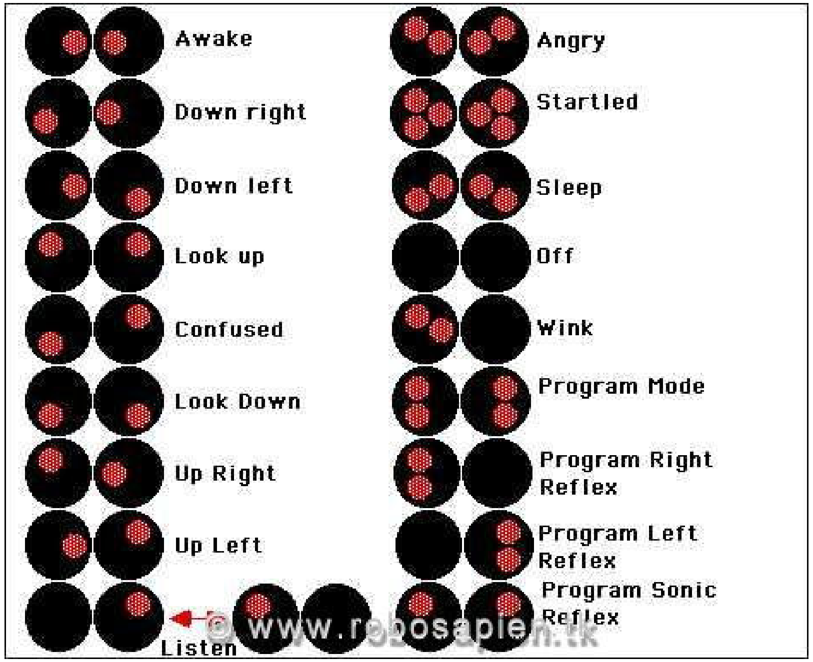
RS's Remote Control

- The RS's remote control unit has 21 different buttons. With the help of two shift buttons, 67 different robot-executable commands are accessible.

Bulldozer	Burp	Right pick up
Fart (Oops)	High Five	Right strike 1
Left arm down	Left arm in	Right strike 2
Left arm out	Left arm up	Right strike 3
Lean back	Lean forward	Right (turn right)
Lean left	Lean right	Step backward
Left pick up	Left strike 1	Step forward
Right throw	Left strike 2	Right sweep
Left sweep	Left strike 3	Right thump
Left thump	Left throw	Roar
Talk back	Walk forward	Whistle
Turn step right	Dance	Demo 1
Walk backward	Demo2	All Demo
Reset	Left (turn left)	Stop
Right arm in	Right arm down	Turn step left
Right arm up	Right arm out	Wake up

RS eye patterns

- Depending on the command that is executed, the RS's eyes have different patterns.
- The 6 outputs for the eyes (P2.0 to P2.5) can be used as a very effective digital-level feedback source. His eyes give distinct patterns for all conditions.



User modifications

- Some words of the Robot Tech Support, from WowWee Ltd.

"The RoboSapien is designed for modification. Here is the short hint list for the budding RS hacker.

First off, we must warn you that completely replacing the RS brain should only be attempted by those with a lot of time, electronic skills, and programming ego.

You don't have to though — if you carefully remove the connectors and lift the RS motherboard, on the back you will find all inputs and outputs labeled, and right next to gold pads convenient for soldering wires."

in http://www.robosapien1.com/resources/official-mod-guide/

User modifications

- Mark Tiden designed the RS to be easily modified or hacked;
- The electronics inside the RS are easily accessed and clearly labeled;
- A growing community of hackers have devoted themselves to adding new functionalities to the robot.

User modifications

- **microbi's Robosapien mods** (http://www.angelfire.com/droid/rsv2/)
- Active mods: hand-beams, hand-Led's, heartbeat, voice off, tunnel-beam, blue EYES

- **Robosapien RF Sound Mod** (http://home.comcast.net/~robosapien/rfmod.htm)
  Robosapien Camera Mod (http://home.comcast.net/~jsamans/robo/robocam.htm)
- Active mods: wireless camera;

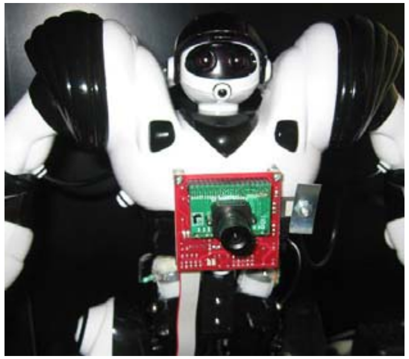
  wireless radio; frequency audio;

  pc control.

Original RoboSapien

User modifications

- **RoboSapienPets RoboSapien page** (http://www.aibohack.com/robosap/)
- Active mods: SuperSapien microcontroller mod, color and motion tracking CMUCam



- **Mark C's Robosapien Hacking Site** (http://homepages.strath.ac.uk/~lau01246/robot/myhackrs.shtml)
- Active mods: microcontrollers (PicMicro controllers, and Palm Pilot controllers for the Robosapien)

Original RoboSapien

Original RoboSapien

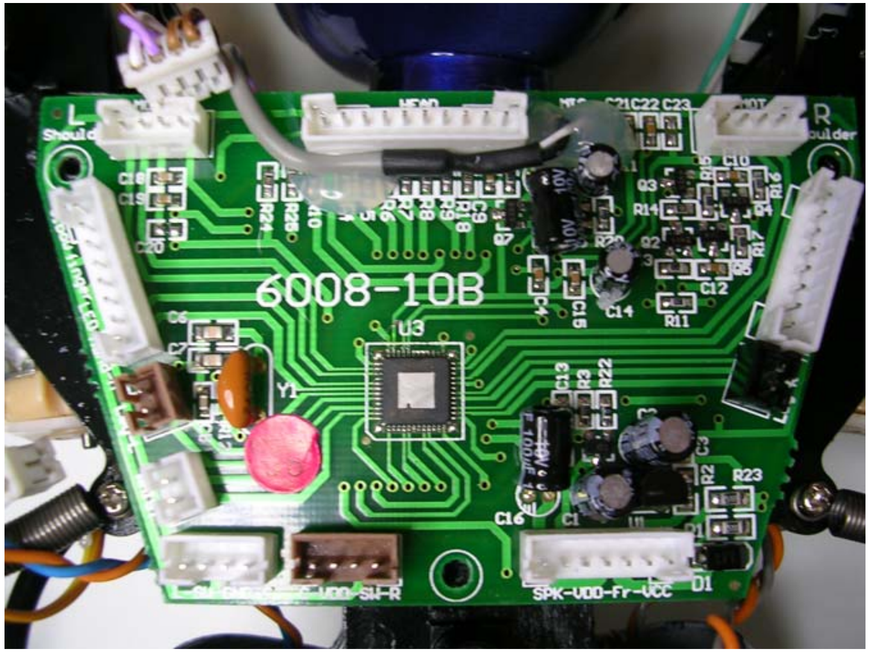
User modifications

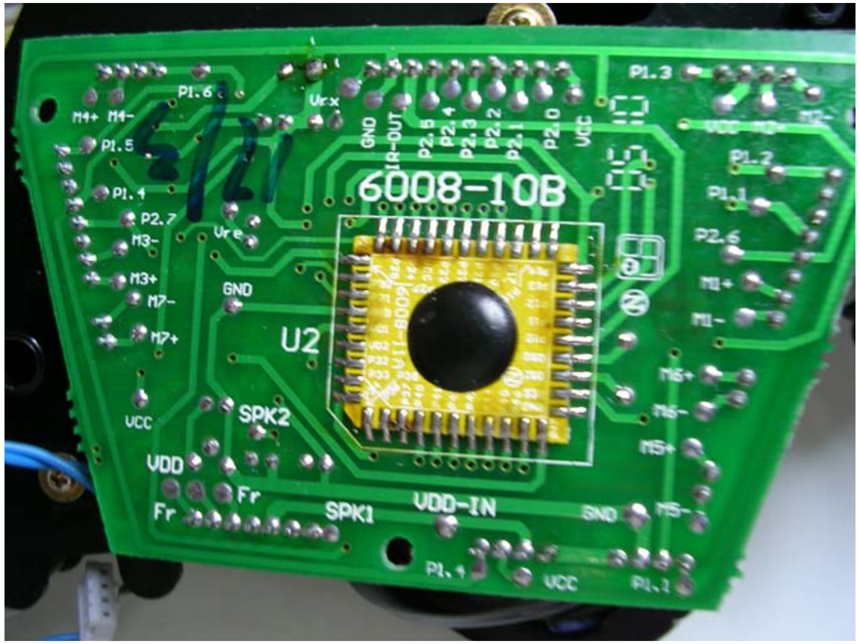
- Robocup German Open 2005 tournament;
- 2 teams of 3 RSs each played the 1st soccer match of humanoid robots worldwide;
- Head replaced by a PDA allowing the perception of its environment using the camera;
- Information sended though the PDA's IR, to a PC.




Controller U2

- **M** means motors,
- **P** means input or output port,
- **VDD** is raw battery voltage (caution: fluctuates wildly)
- **Vcc** is regulated 3.6V and **Gnd** is universal ground.





Controller U2 - Connections

Original RoboSapien

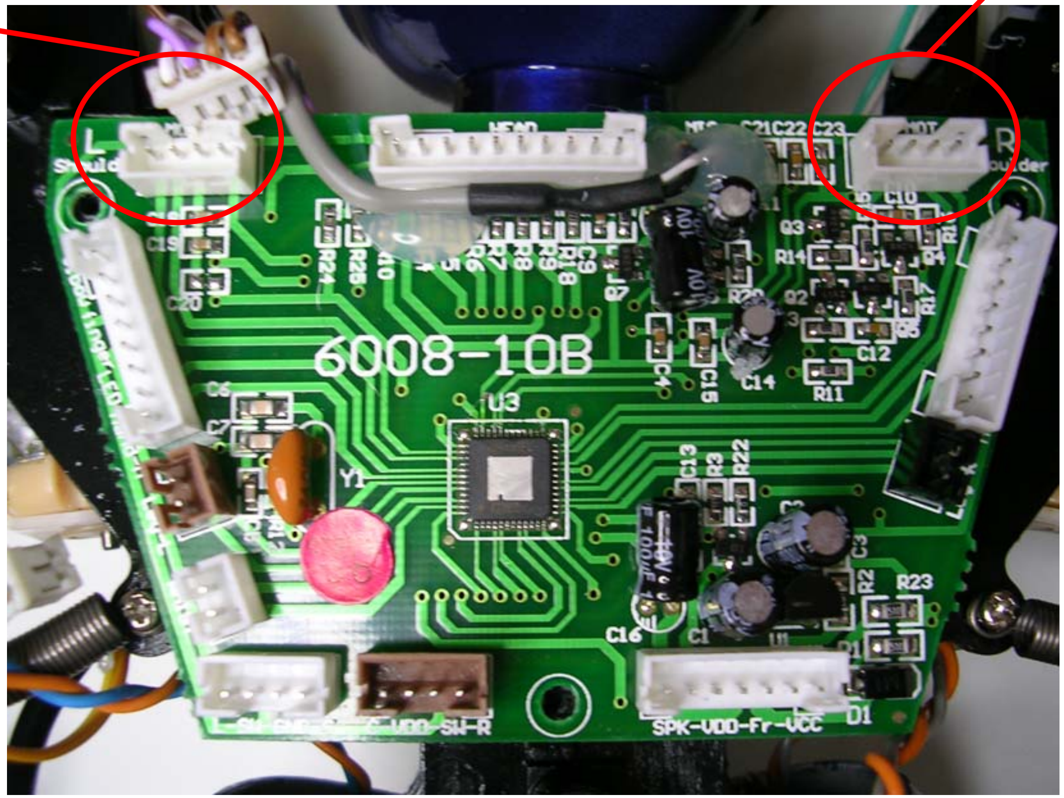
Left shoulder

M2- M2+ Vcc P1.3

Motors

Right Shoulder

P1.6 Vcc M4- M4+

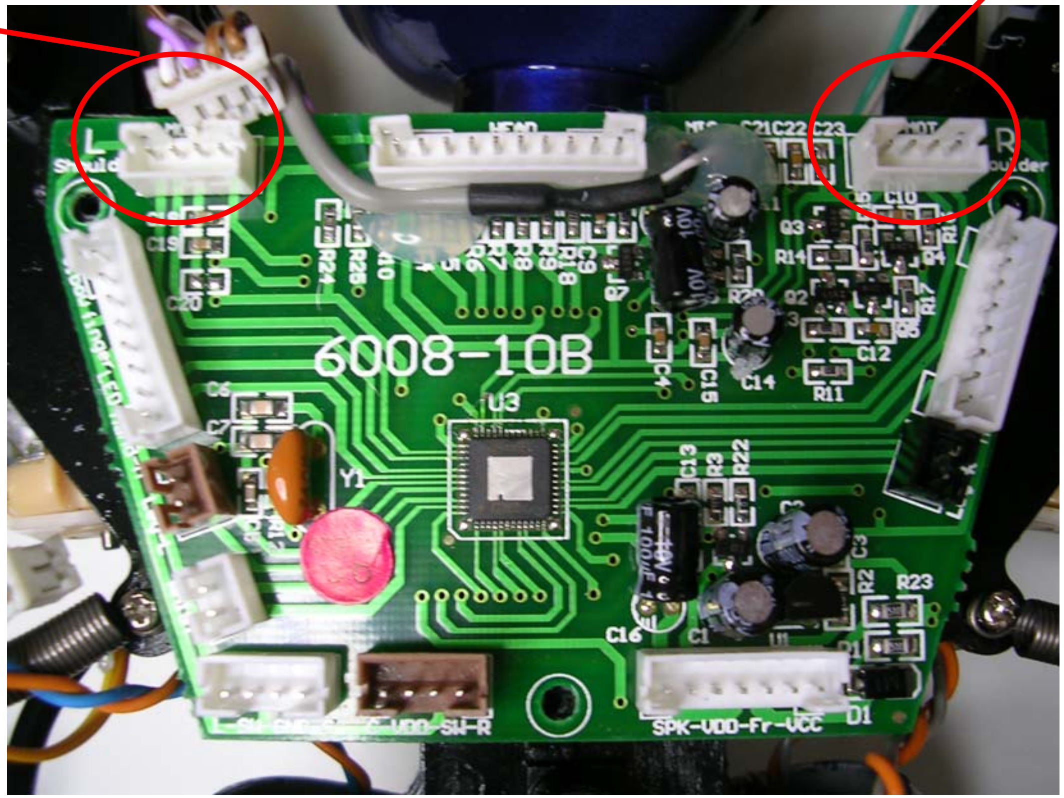


Controller U2 - Connections

Left shoulder — **Shoulder position switch** — **Right Shoulder**

M2- M2+ Vcc P1.3

P1.6 Vcc M4- M4+



Controller U2 - Connections

Original RoboSapien

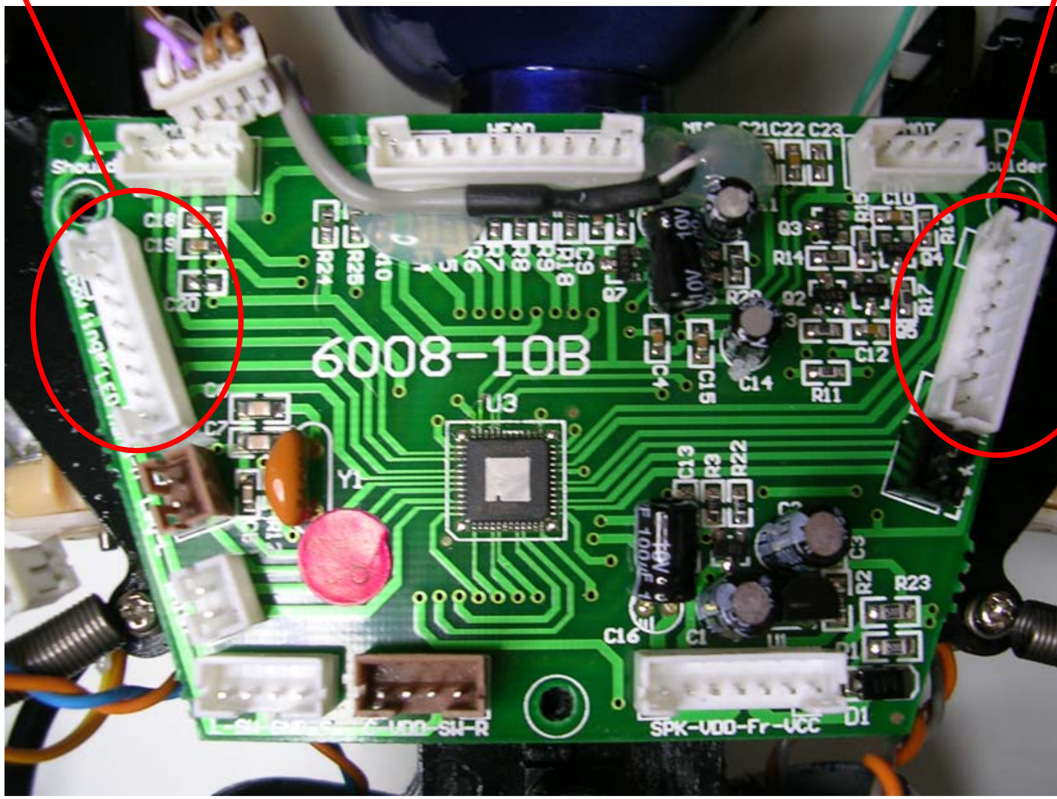
Motors

Left elbow

- P1.2
- Vcc
- Vcc
- P1.1
- P2.6
- Vcc
- M1+
- M1-

Right elbow

- M3+
- M3-
- Vcc
- P2.7
- P1.4
- Vcc
- Vcc
- P1.5

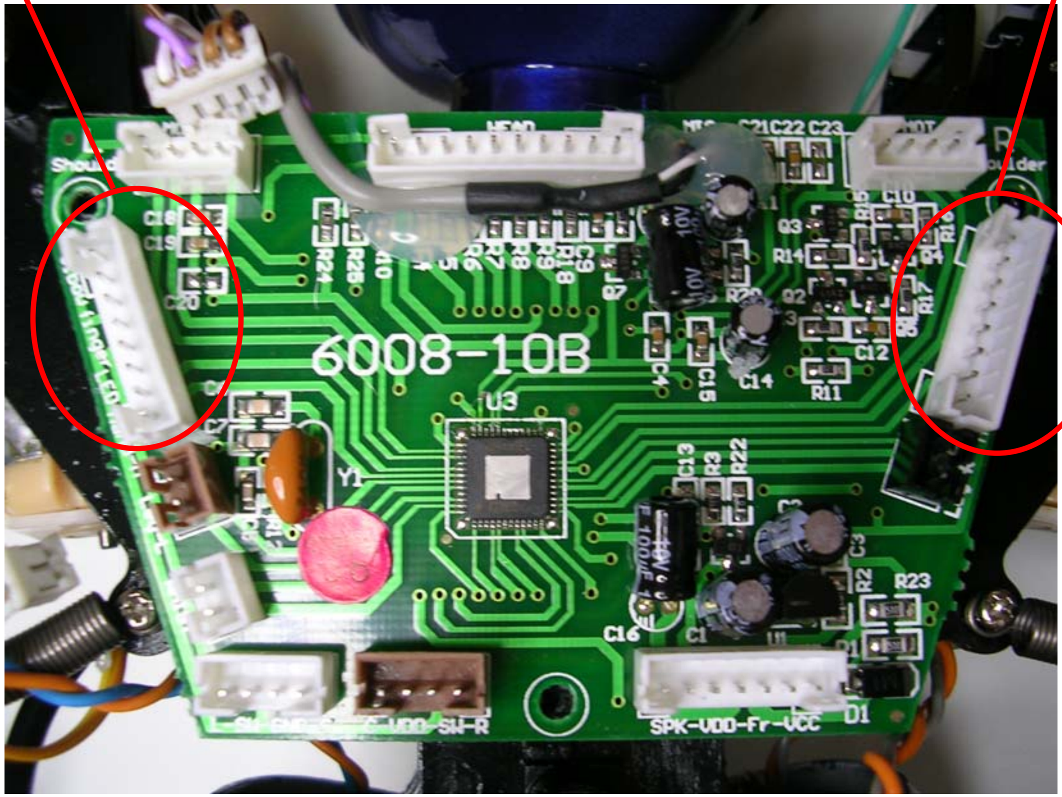


Controller U2 - Connections

Left elbow

P1.2
Vcc
Vcc
P1.1
P2.6
Vcc
M1+
M1-

Finger switch



Right elbow

M3+
M3-
Vcc
P2.7
P1.4
Vcc
Vcc
P1.5

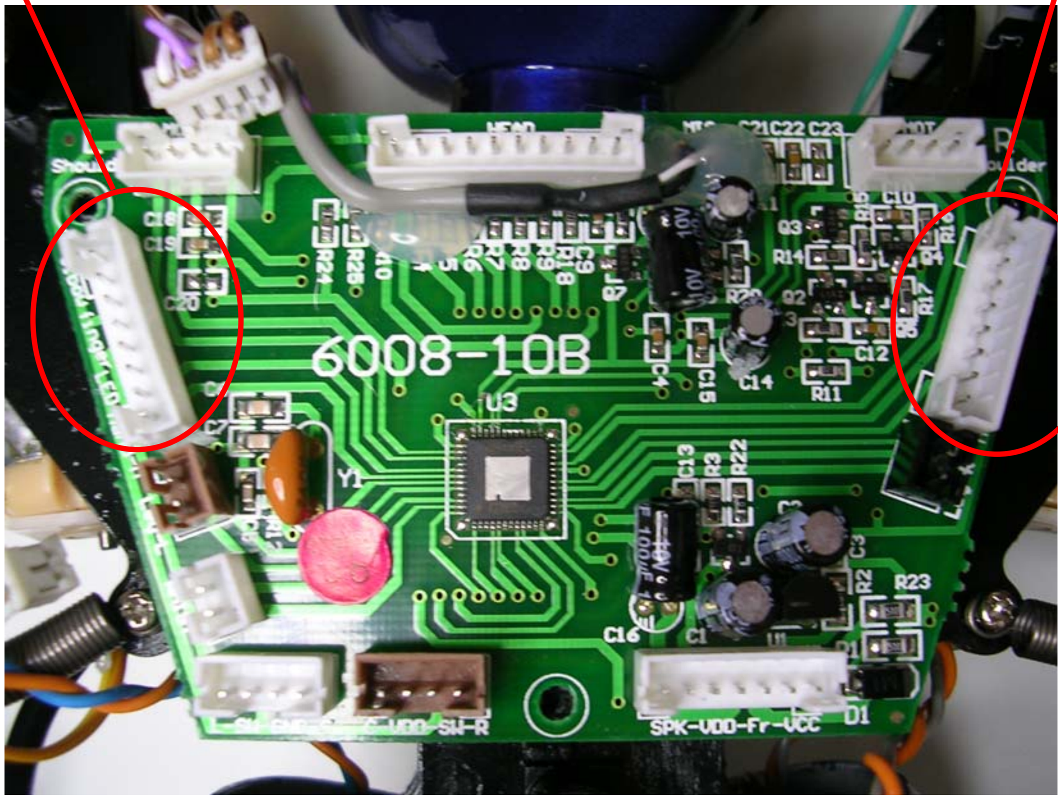
Original RoboSapien

Controller U2 - Connections

Left elbow

P1.2
Vcc
Vcc
P1.1
P2.6
Vcc
M1+
M1-

Elbow position switch



Right elbow

M3+
M3-
Vcc
P2.7
P1.4
Vcc
Vcc
P1.5

Controller U2 - Connections

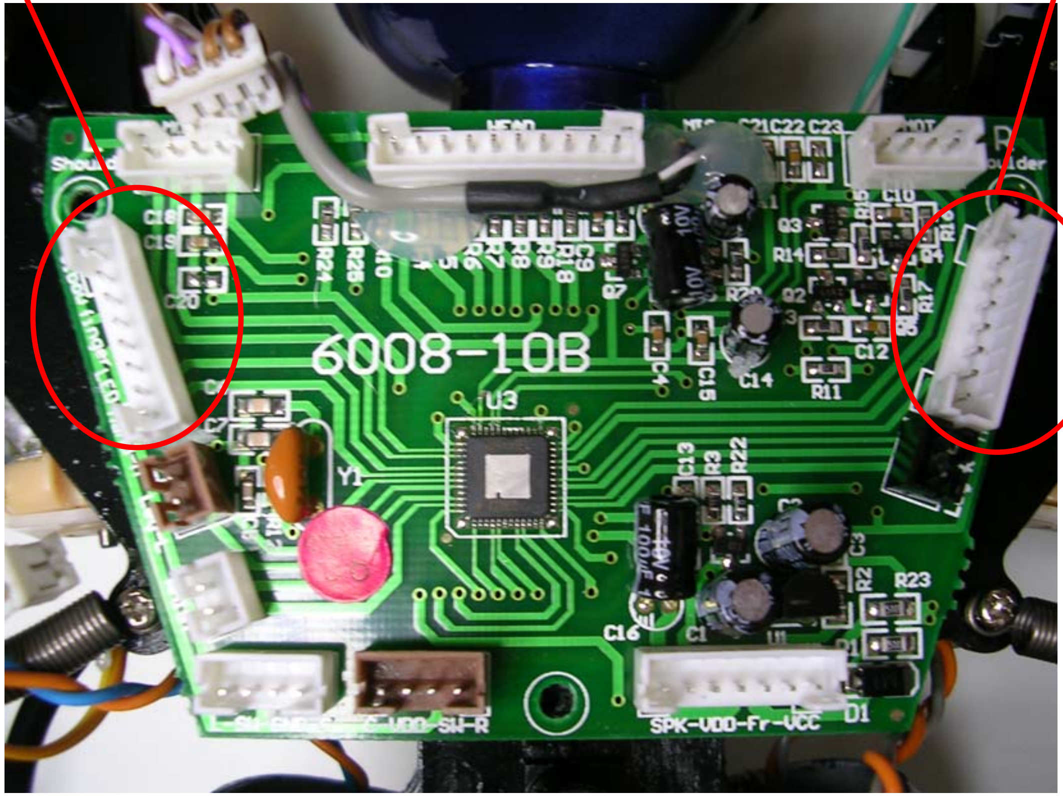
Left elbow

P1.2
Vcc
Vcc
P1.1
P2.6
Vcc
M1+
M1-

Arm LEDs

Right elbow

P1.5
Vcc
Vcc
P1.4
P2.7
Vcc
M3-
M3+



Controller U2 - Connections

Left leg/hip

M6+

M6-

Motors

Right leg/hip

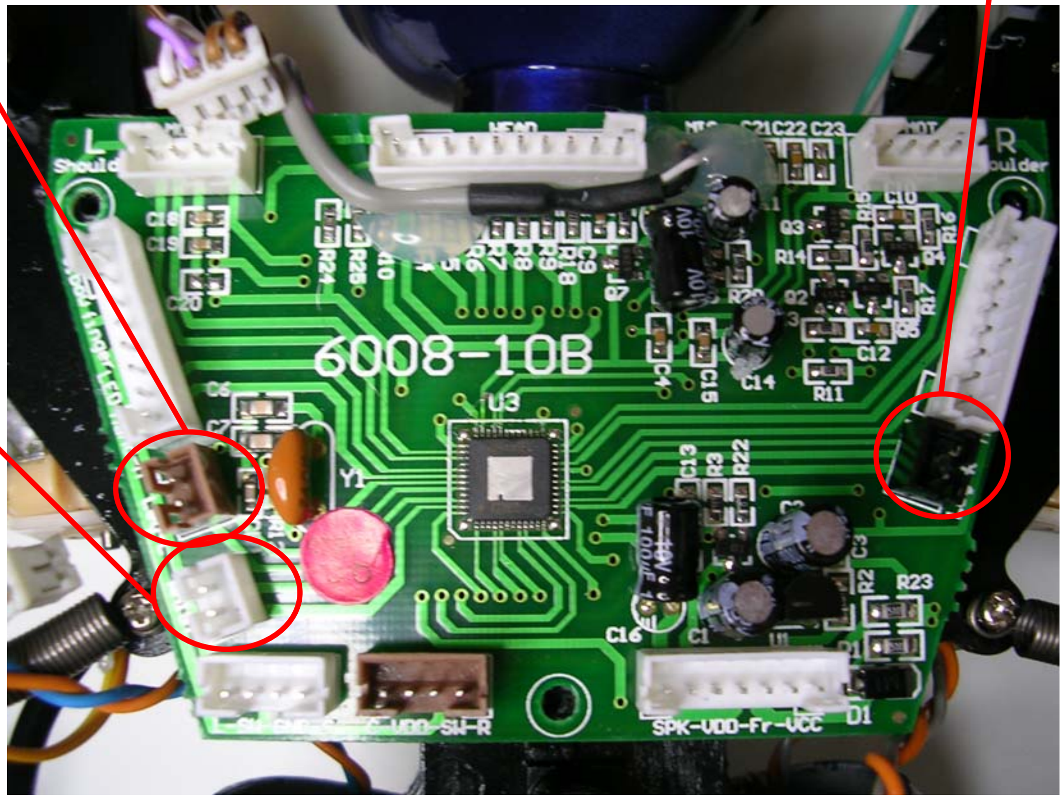
M7-

M7+

Tilt

M5+

M5-

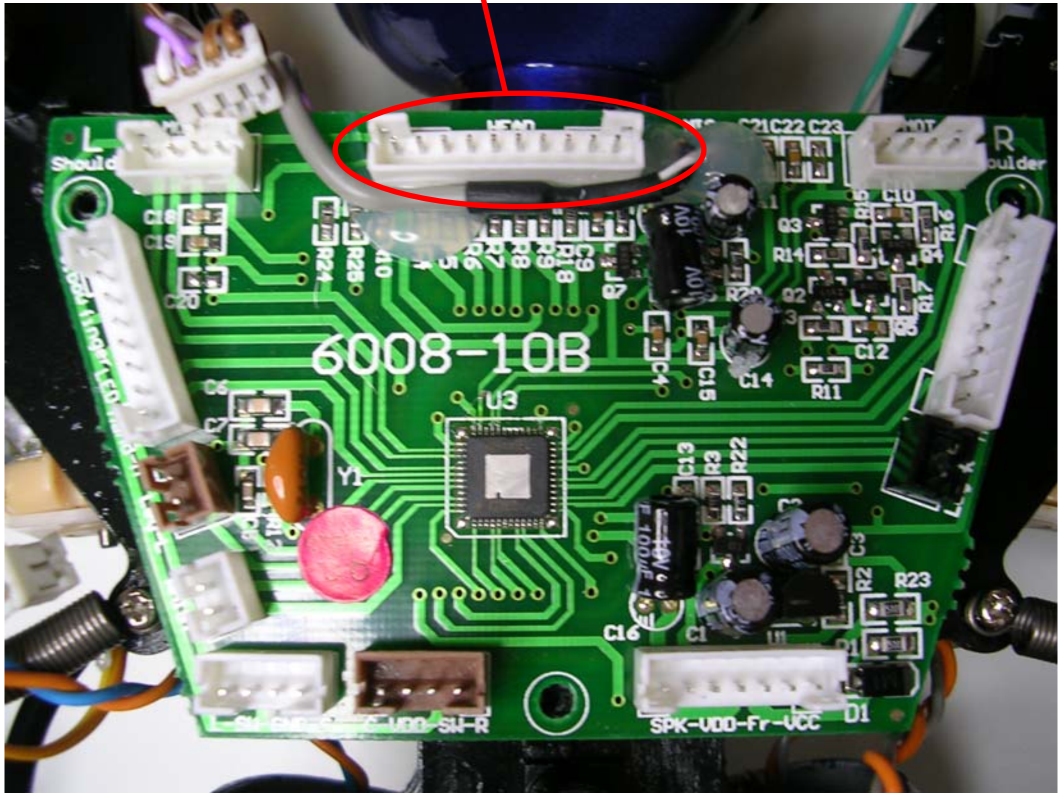


Controller U2 - Connections

Head

Vcc P2.0 P2.1 P2.2 P2.3 P2.4 P2.5 IR-OUT Gnd Vrx

Left eye LEDs

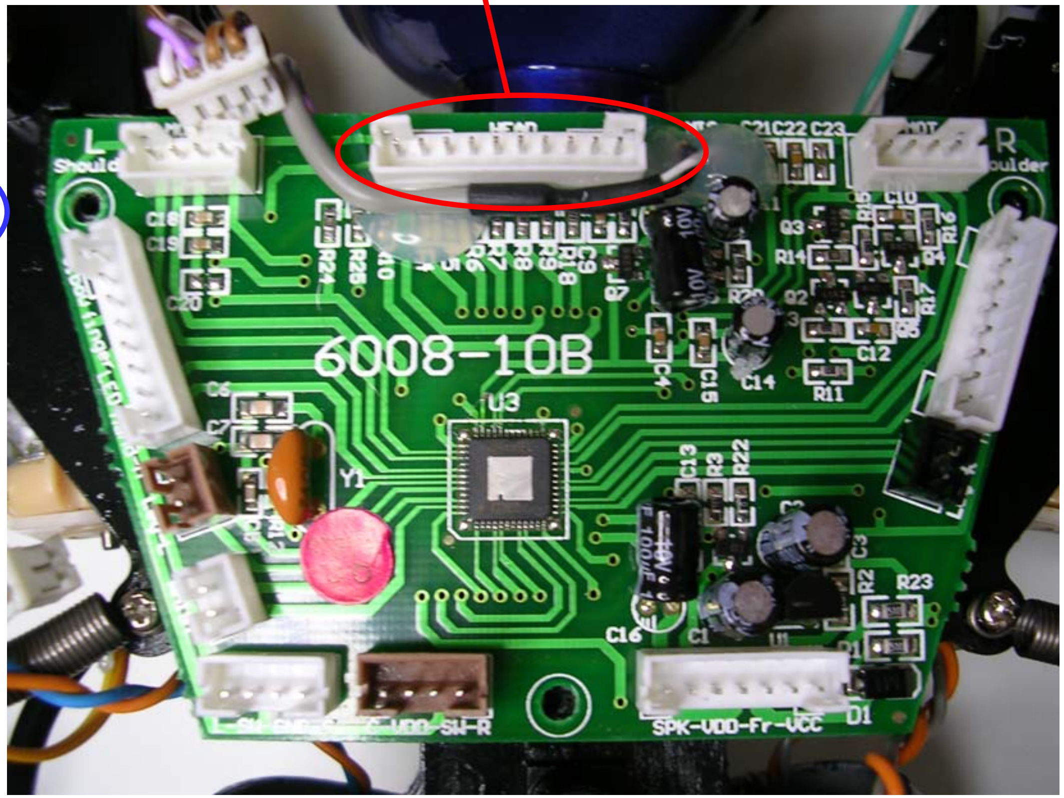


Controller U2 - Connections

Head

Vcc P2.0 P2.1 P2.2 P2.3 P2.4 P2.5 IR-OUT Gnd Vrx

Right eye LEDs



Controller U2 - Connections

Component

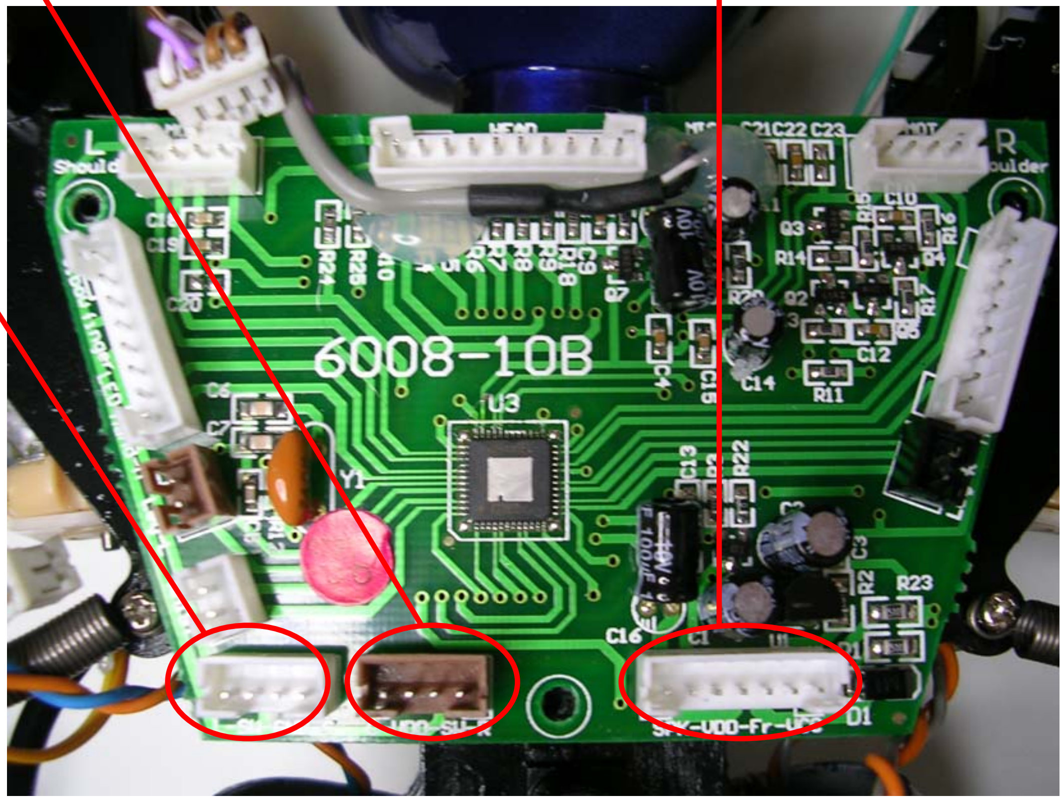
Vcc VDD Vcc P1.4

Speakers

SPK1 SPK1 IN OUT Fr VDD Fr

Component

P1.1 Vcc Gnd Vcc



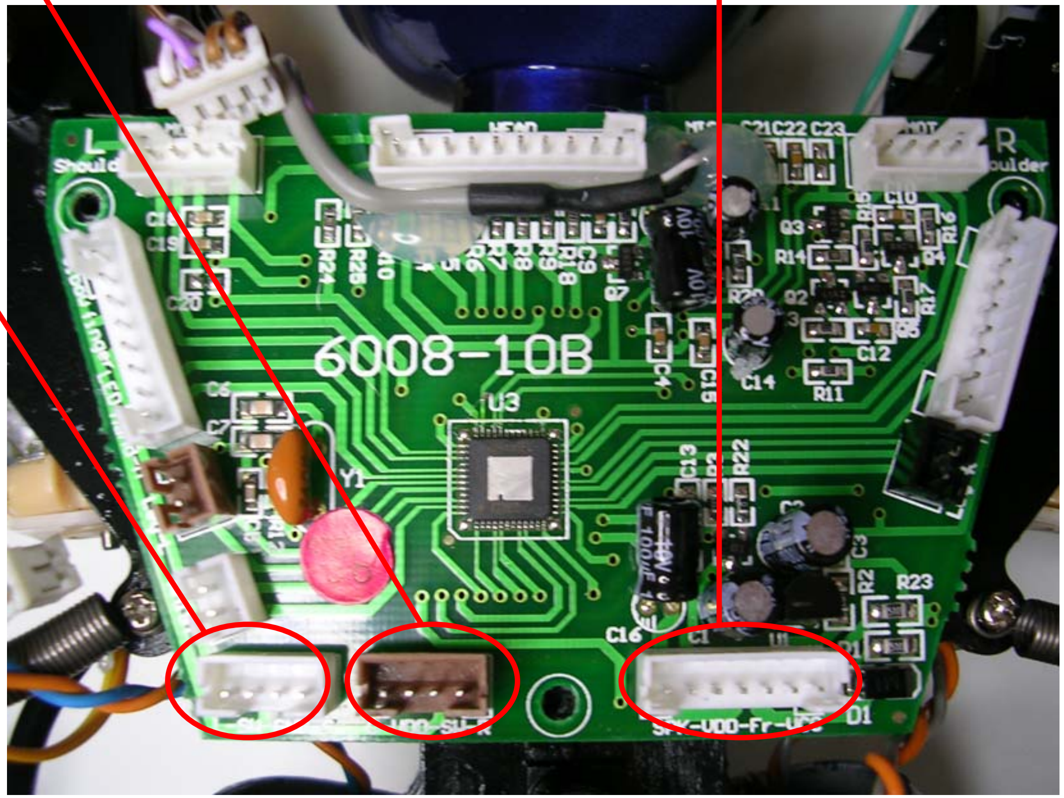
Controller U2 - Connections

Component **Foot switch** **Speakers**

Vcc VDD Vcc P1.4 SPK1 SPK1 IN OUT Fr VDD Fr

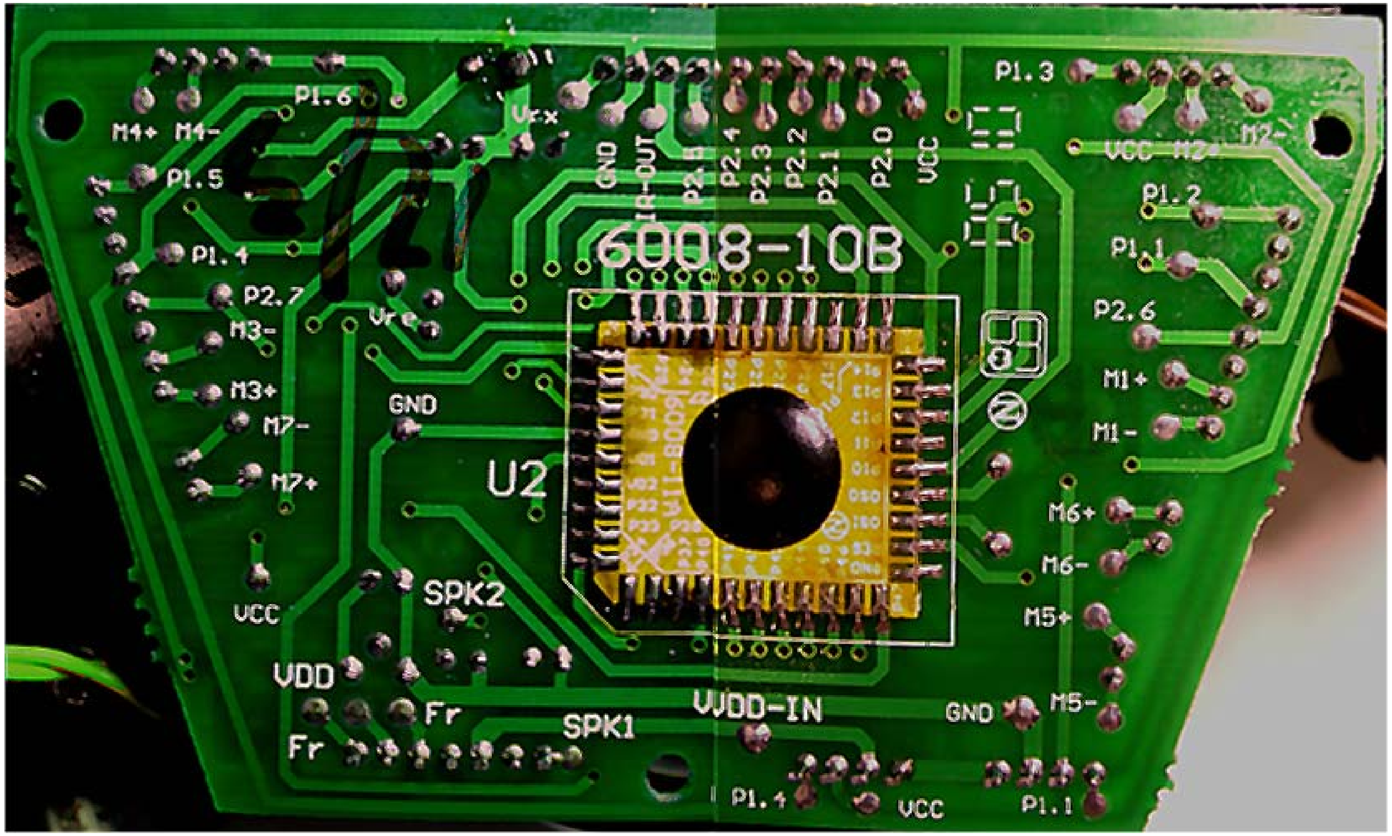
Component

P1.1 Vcc Gnd Vcc

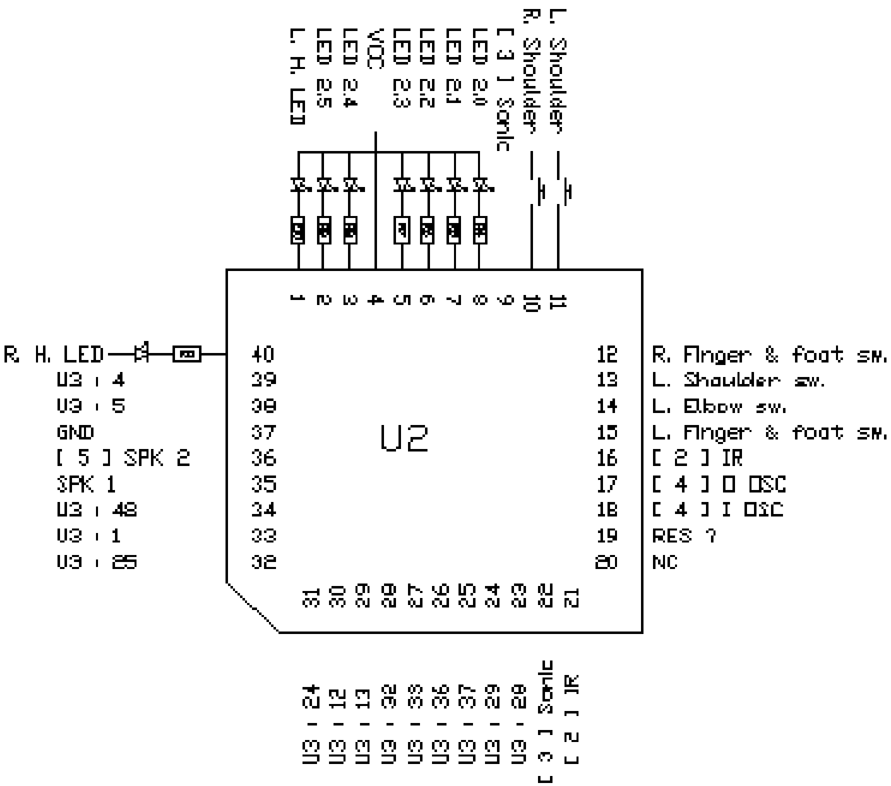


Original RoboSapien

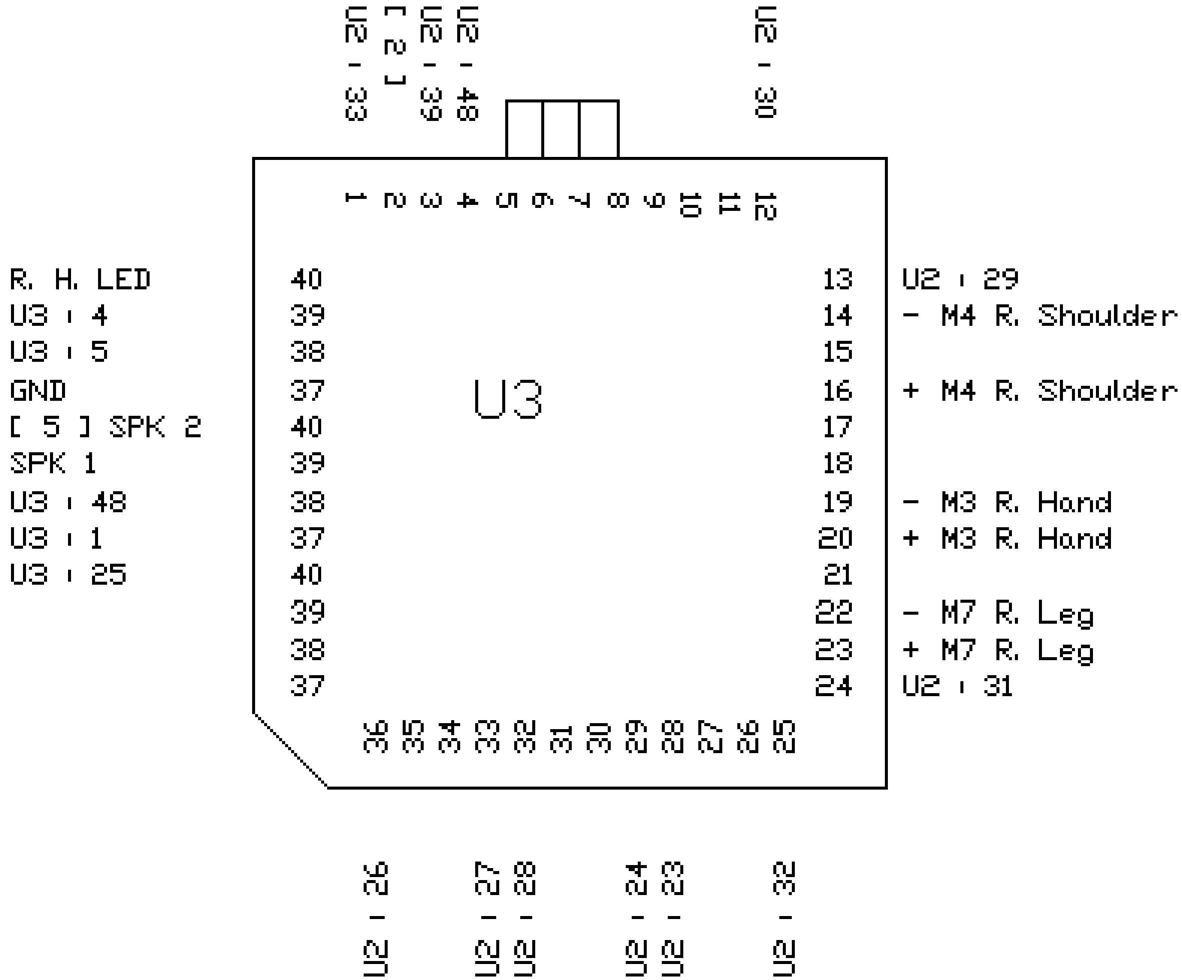
Controller U2



Controller U2



Motor driver U3

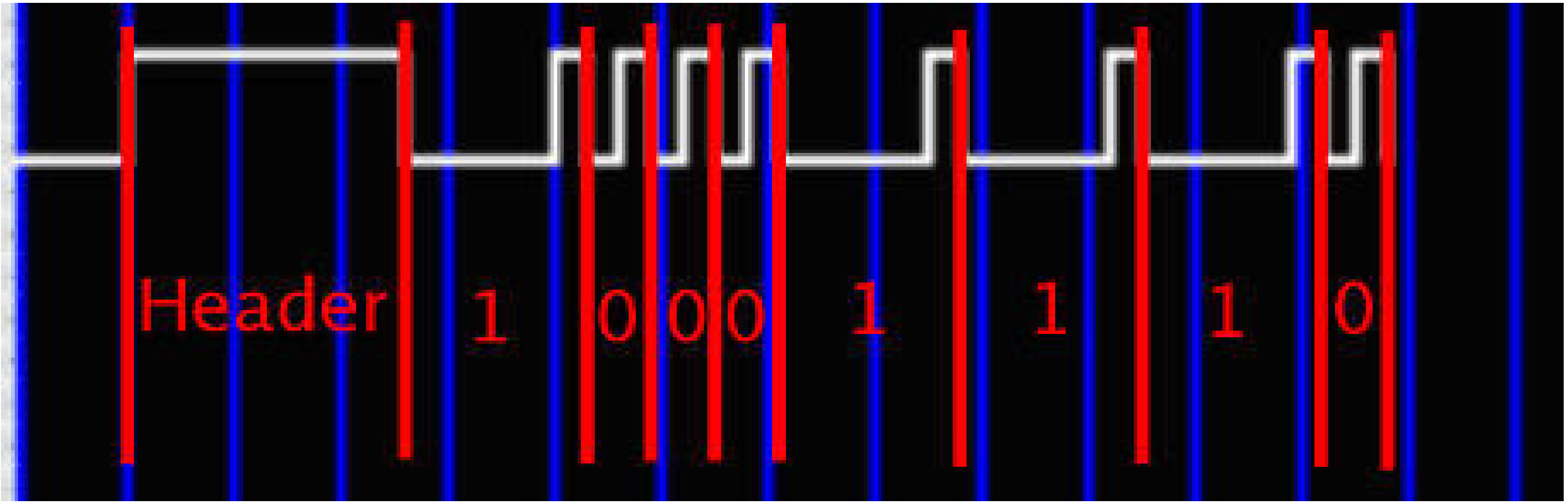


IR commands

- To input commands is used the direct serial input to the IR-OUT pin (active low signals, 1200 bps).
- Timing based on 1/1200 second clock (~ 0,833 msec) Signal is normally high (idle, no IR).
- Data bits: for each of 8 data bits, space encoded signal depending on bit value (Sends the most significant data bit first). (Carrier is 39,2 kHz)
- BTW: The first bit (msb) is always 1 (valid codes are from 0x80 to 0xFF)

IR commands

- Header: signal goes low for 8/1200 sec.
- If the data bit is 0: signal goes high for 1/1200 sec, and low for 1/1200 sec.
- If the data bit is 1: signal goes high for 4/1200 sec, and low for 1/1200 sec.
- Example: Command Stop: 0x8E



Original RoboSapien

RoboSapien Replication

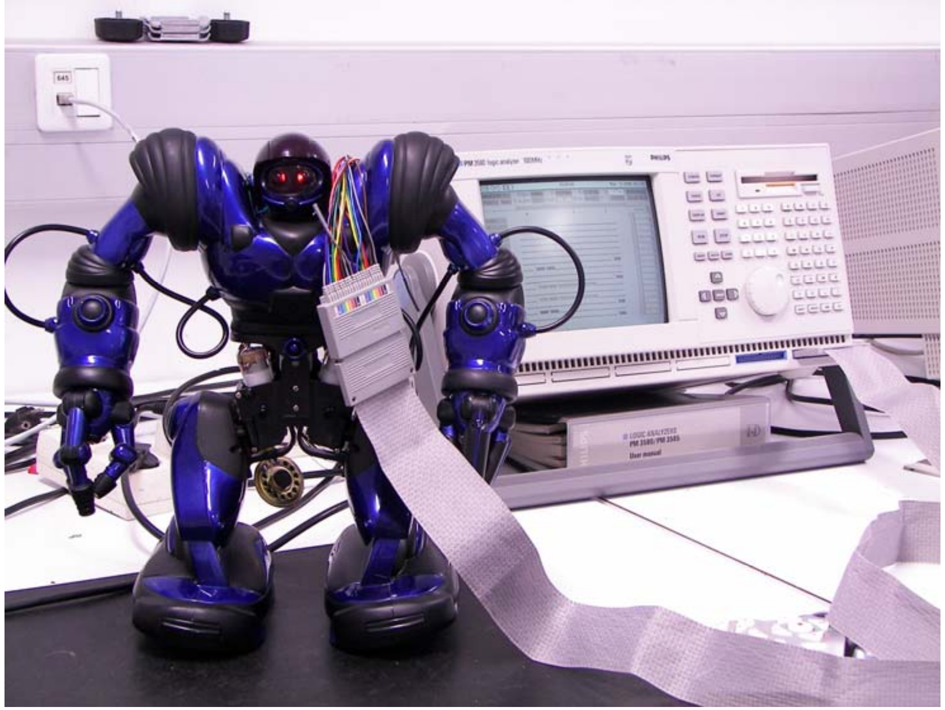
- The RS has the following devices and sensors:
  - IR receiver;
  - Power switch;
  - Microcontroller;
  - Seven motors (hands, arms, feet and hip);
  - Four foot touch sensors;
  - Two hand touch sensors,
  - Sound sensor;
  - Seven LEDs;
  - An external IR remote control.

Signal Analysis

- Digital signals acquired from the electronic board ports;
- To know how the microcontroller work when the robot do a specific command function.
- The procedure consisted in the acquisition of the ports digital signals:
  - for a single motor,
  - related to command functions that combine several movements at the same time.

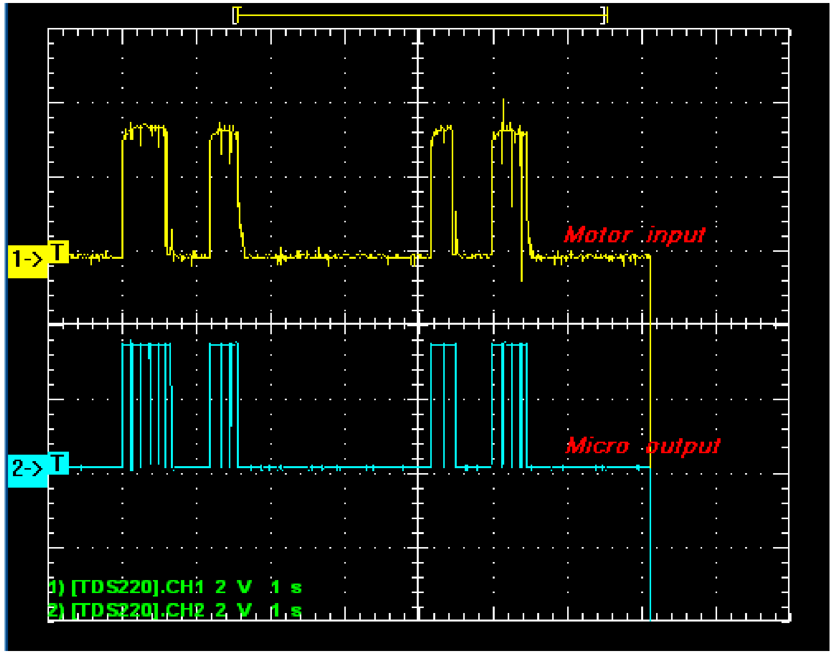
Signal Analysis

- For the single movements we've used a Tektronix™ TDS220 oscilloscope.
- For the robot combined movements, a PM3580 Logic Analyzer was used.

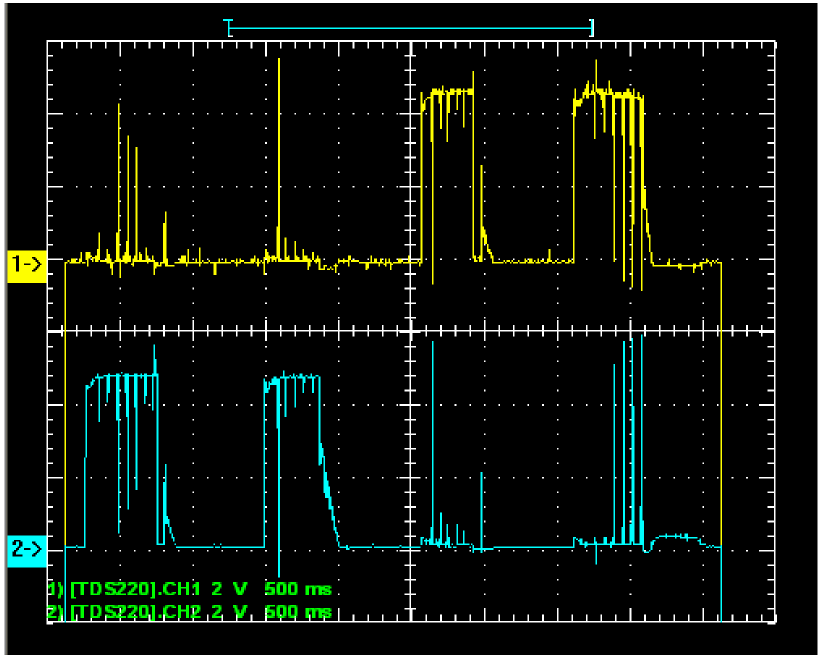


Single motor signal graphics

- Illustration of the output signal from the original microcontroller and the signal that the motor receives.



output signal vs. Motor input signal



Left elbow movement

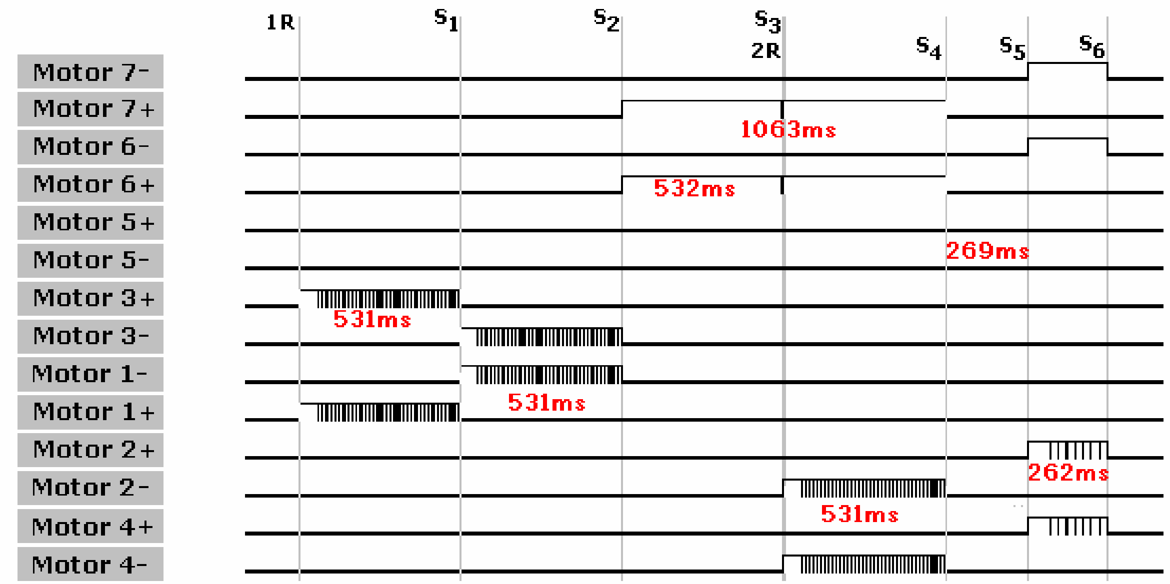
from the inside to outside and vice-versa

Combined motors graphics

- Wires connected to the original microcontroller ports:
  - In order to connect it to a flat cable to measure the combined movements digital signals using a logic analyzer.
- Acquisition of the graphical digital signals for the microcontroller motor ports:
  - For all the combined functions defined in the remote control.
- Graphic functions only obtained for a side function movement of the robot:
  - Since other side does the same movements with opposite motors.

Combined motors graphics

- Example: Output Ports Signals for the function “Oops”.
- Signal “M1+” (Left Elbow Out) is “high”, for 531 msec and the rest of the time (2125 msec) is “low”.
- Signal “M3+” (Right Elbow Out) presents the same signal.
- Both elbows executes the movement at the same time.

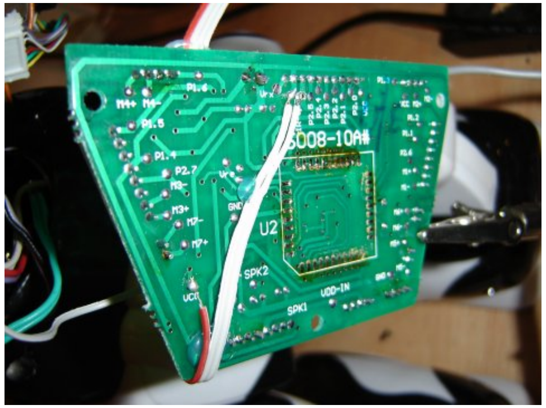


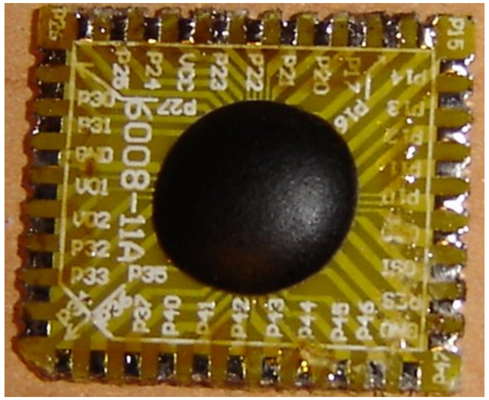
Electromechanical engineering classes - Robotics

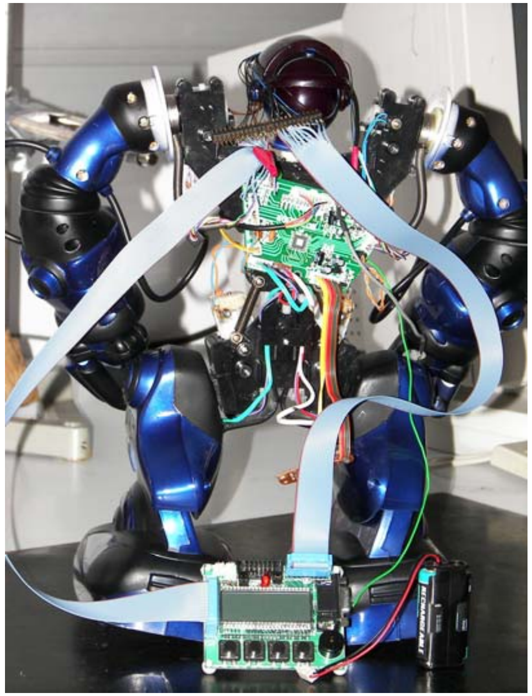
Electromechanical engineering classes - Robotics

New RS Brain – MSP430

- Removal of the original microcontroller from the RS electronic board.







- Flat cables were used to wire the electronic board to the kit Olimex MSP430F449-STK2 to test the motors.

MSP430 I/O functions

Label	MSP430F449 Ports	Motor	Function
M1+	P6.0	Left Elbow	Elbow goes out, and opens gripper
M1-	P6.1	Left Elbow	Elbow goes in, and closes gripper
M2+	P6.2	Left Arm	Arm goes up
M2-	P6.3	Left Arm	Arm goes down
M3+	P6.4	Right Elbow	Elbow goes out, and opens gripper
M3-	P6.5	Right Elbow	Elbow goes in, and closes gripper
M4+	P6.6	Right Arm	Arm goes up
M4-	P6.7	Right Arm	Arm goes down
M5+	P2.0	Hip	Left leaning
M5-	P2.1	Hip	Right leaning
M6+	P2.2	Left Foot	Foot goes back
M6-	P2.3	Let Foot	Foot goes forward
M7+	P2.4	Right Foot	Foot goes back
M7-	P2.5	Right Foot	Foot goes forward

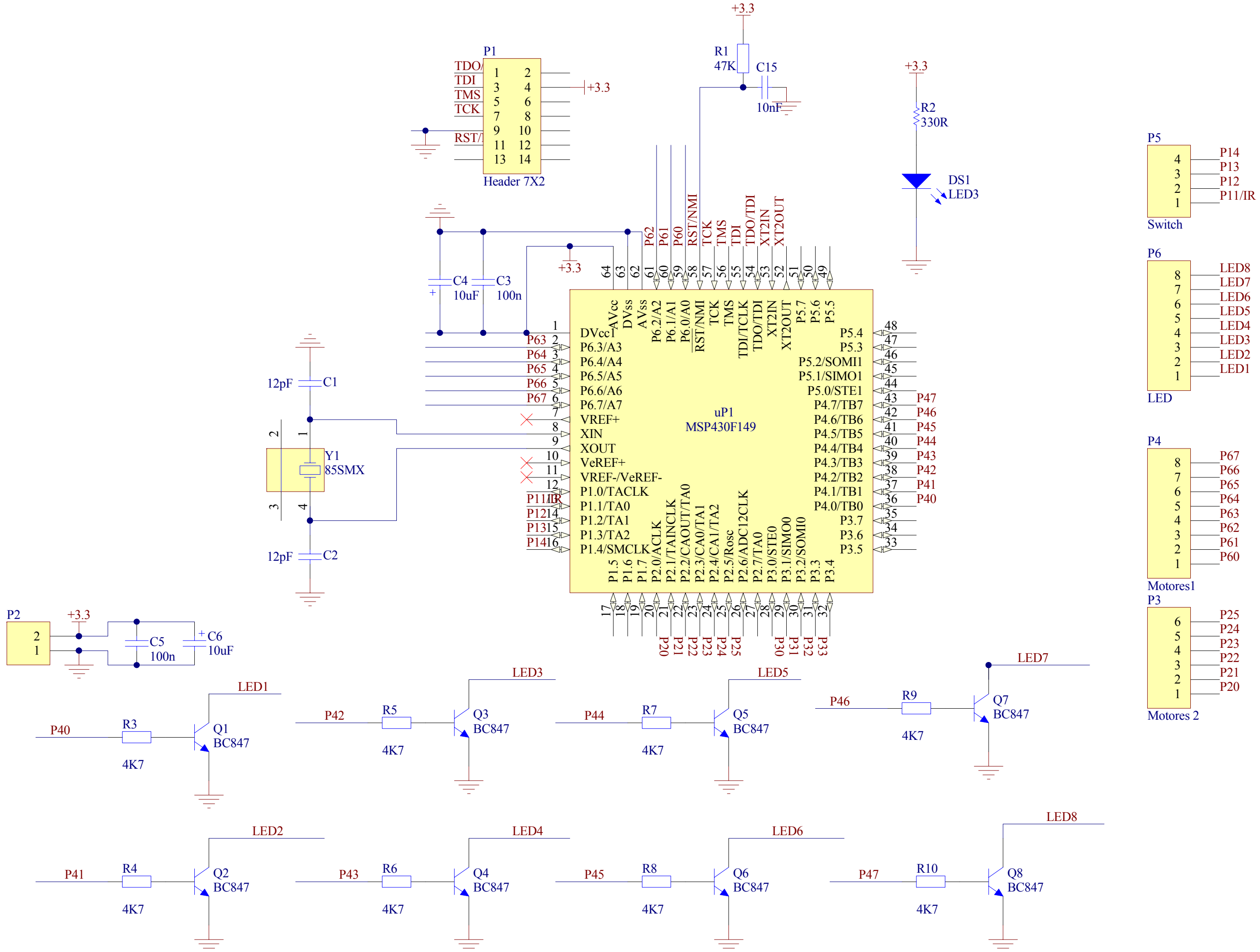
Hardware - PCB board

- Microcontroller: MSP430F149;

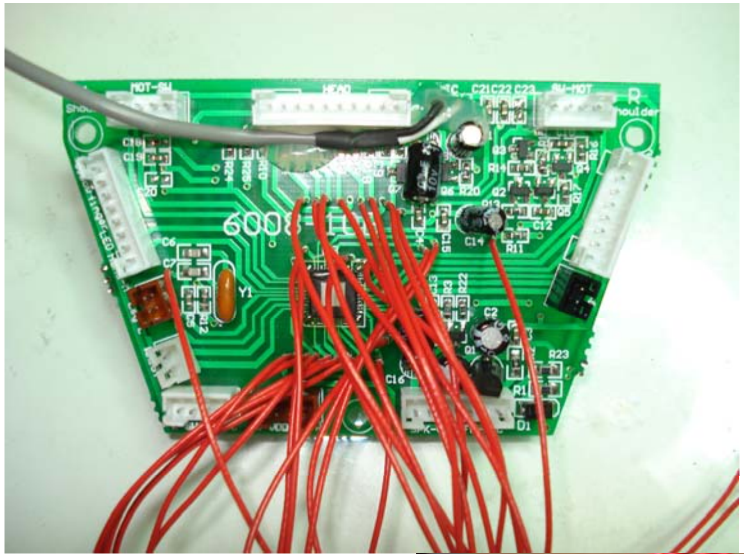
- Resources:
  - Motors: P6.0 – P6.7 , P2.0 – P2.5;
  - LEDs: P4.0 – P4.7;
  - IR: P1.1;
  - Switches: P1.2 – P1.3;

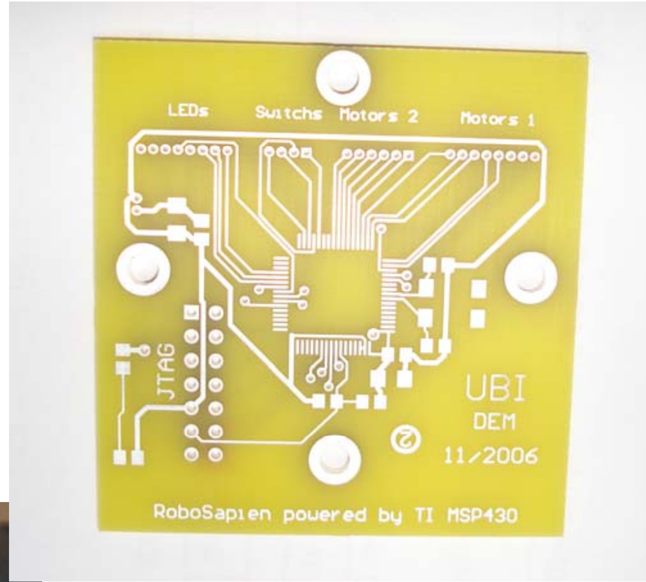
Electromechanical engineering classes - Robotics

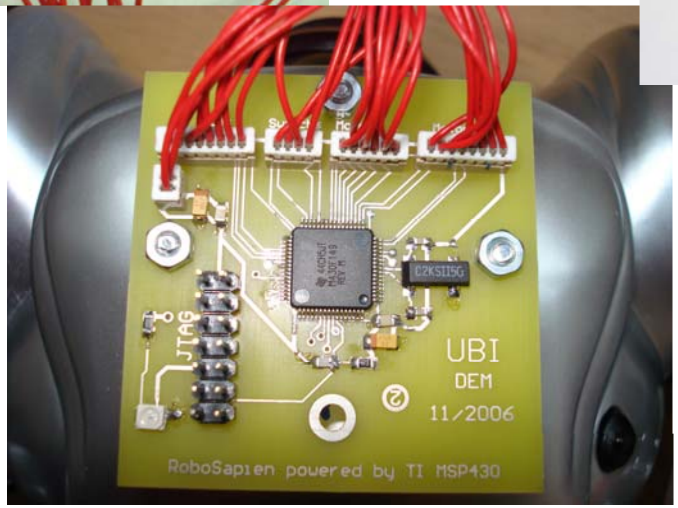
Hardware - PCB board



Hardware - PCB board







Software – Operations Principle

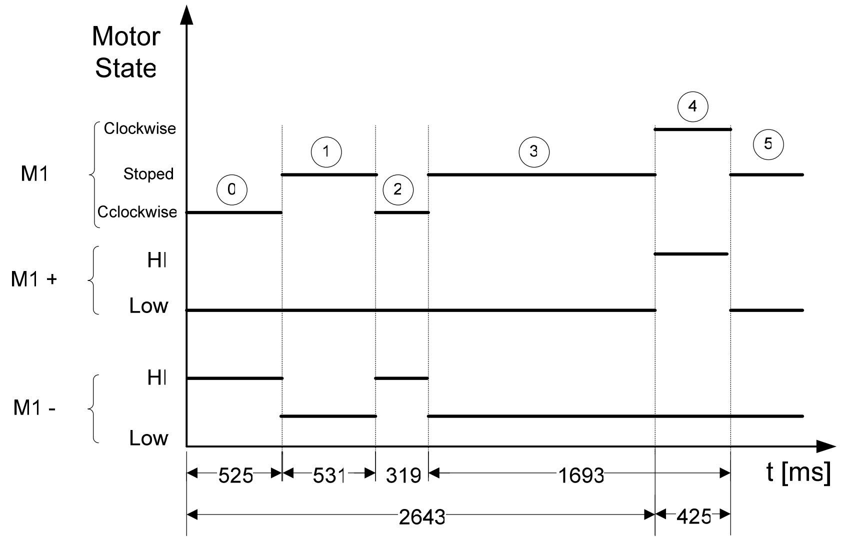
Timers

Motors		[0]	[1]	[2]	[3]	[4]	...	[13]	[14]	Motor State	Motor Initial Value
Motor 1	M1+ [0]	2643	425	0	0	0		0	0	1	0
	M1- [1]	525	531	319	1693	0		0	0	1	1
	[2]	0	0	0	0	0		0	0	0	0
	...										
	[12]	0	0	0	0	0		0	0	0	0
	[13]	0	0	0	0	0		0	0	0	0

Example: M1 state 0

If M1+ = High & M1- = Low

then, M1 runs counter clockwise

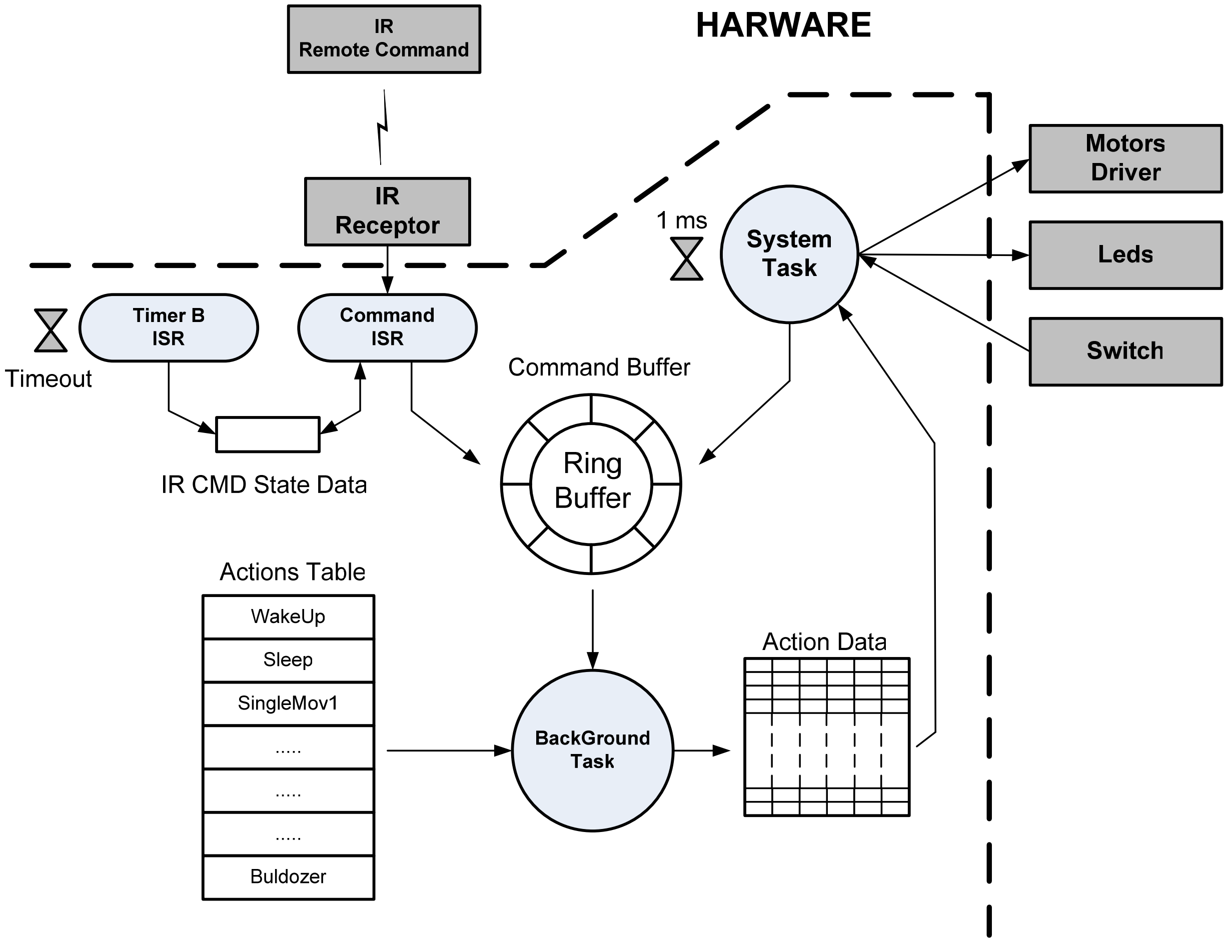


M1+, M1- are logical motors;

Both represent the physical motor M1;

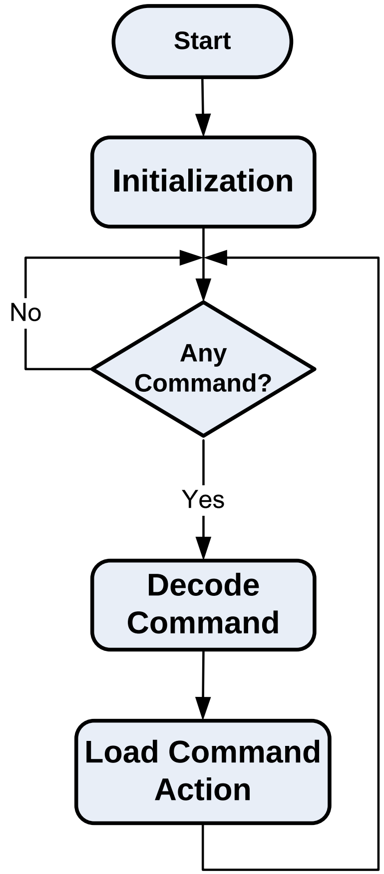
Note: M1+, M1- can't possess the same high state (short circuit)

Software – Architecture



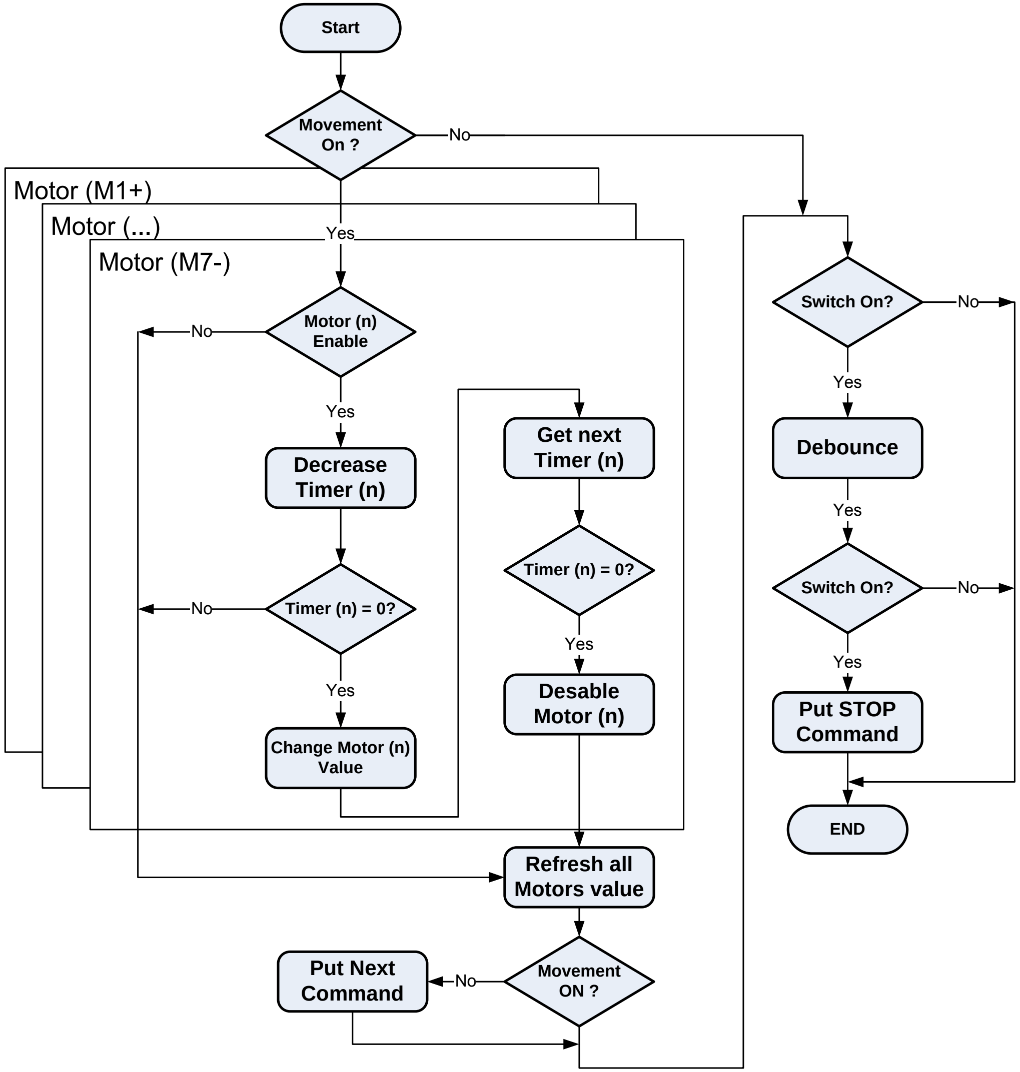
Software – Background task

BackGround Task Flow Chart

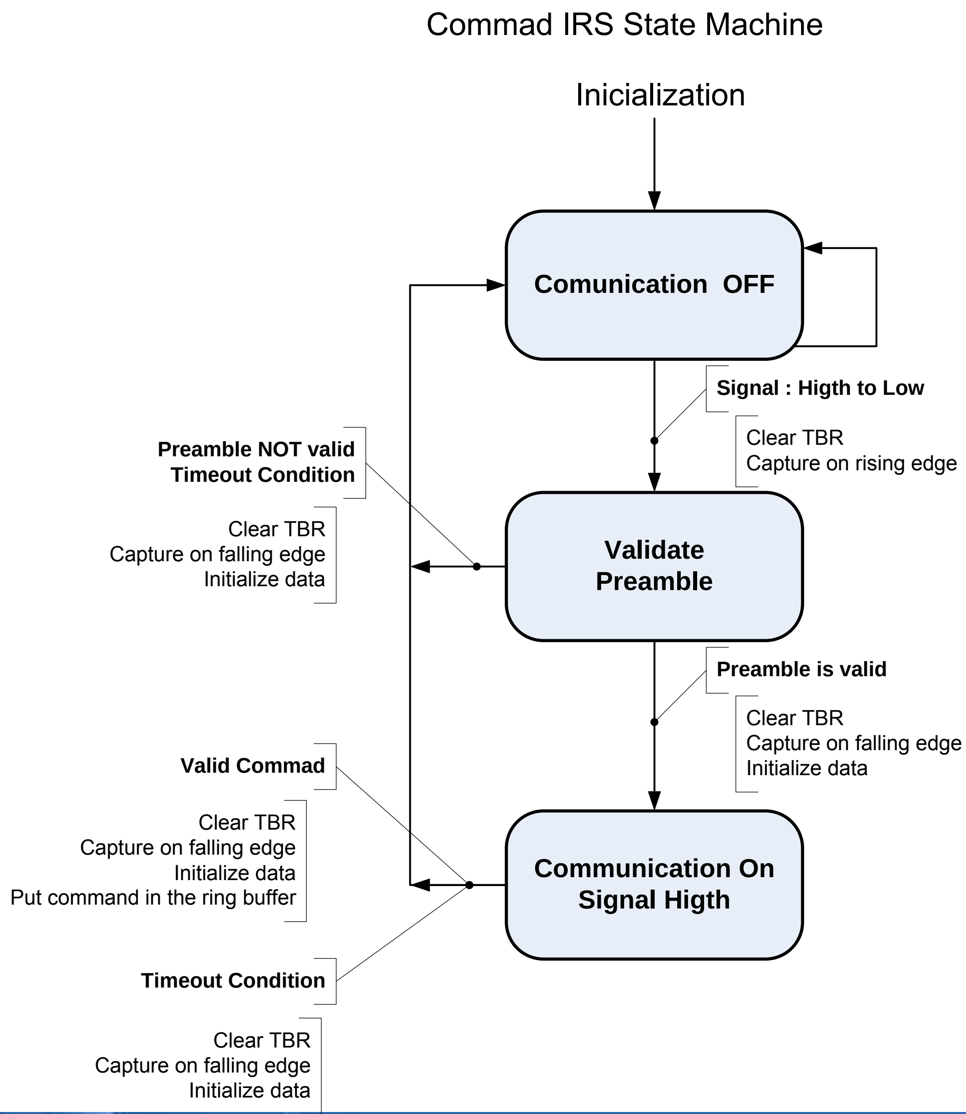


Software – System task

System Task Flow Chart



Software – IR command task



Project conclusions

- Successful attempt in the substitution of RoboSapien control/regulation electronics by TI MSP430;
- Excellent demo vehicle of technology;
- Use the RoboSapien in laboratorial classes, as a way to demonstrate the embedded systems capabilities;
- With MSP430, the RoboSapien has a potential evolution capability.

Future work

- Include wireless communications like ZigBee;
- Expand computation capabilities (TI C2000) to include voice commands;
- Development of a user friendly PC high level application to perform new actions/movements;

MSP430@UBI-DEM

- **Last year projects**

- **Sounds Recording and Reproduction System based in MSP430 for the RoboSapien**

- Implementation of a system of sounds with the particularity to reproduce the voice signal data, stored in *flash* memory by:
- PWM outputs available in the MSP430F449.
- A digital to analog converter TI TLV5616CP.
- The comparison between the methods give clear advantage to the system that uses the external integrated circuit.

PWM frequency

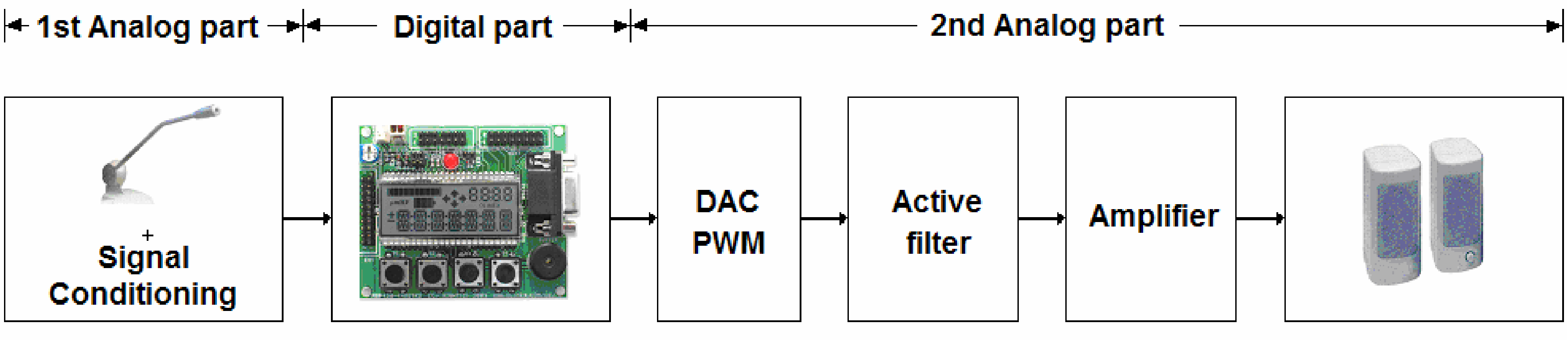
- The required frequency for the PWM output signal is equal to the DAC actualization frequency.

- Each alteration in the PWM *duty cycle* is equivalent to one DAC sampling.

- Clock frequency: 8 MHz
- Resolution: 12 bits
- PWM frequency: 1953 Hz

Global Architecture of the system

- Application block diagram (Analog and digital blocks) since the microphone to the speaker.



MSP430 used peripherals

- MSP430 Digital peripherals:
- ADC12,
- *Timer_A* and *Timer_B*,
- flash memory.

- The signal conditioning circuit of the microphone is connected to channel 0 of the ADC12.

Important States

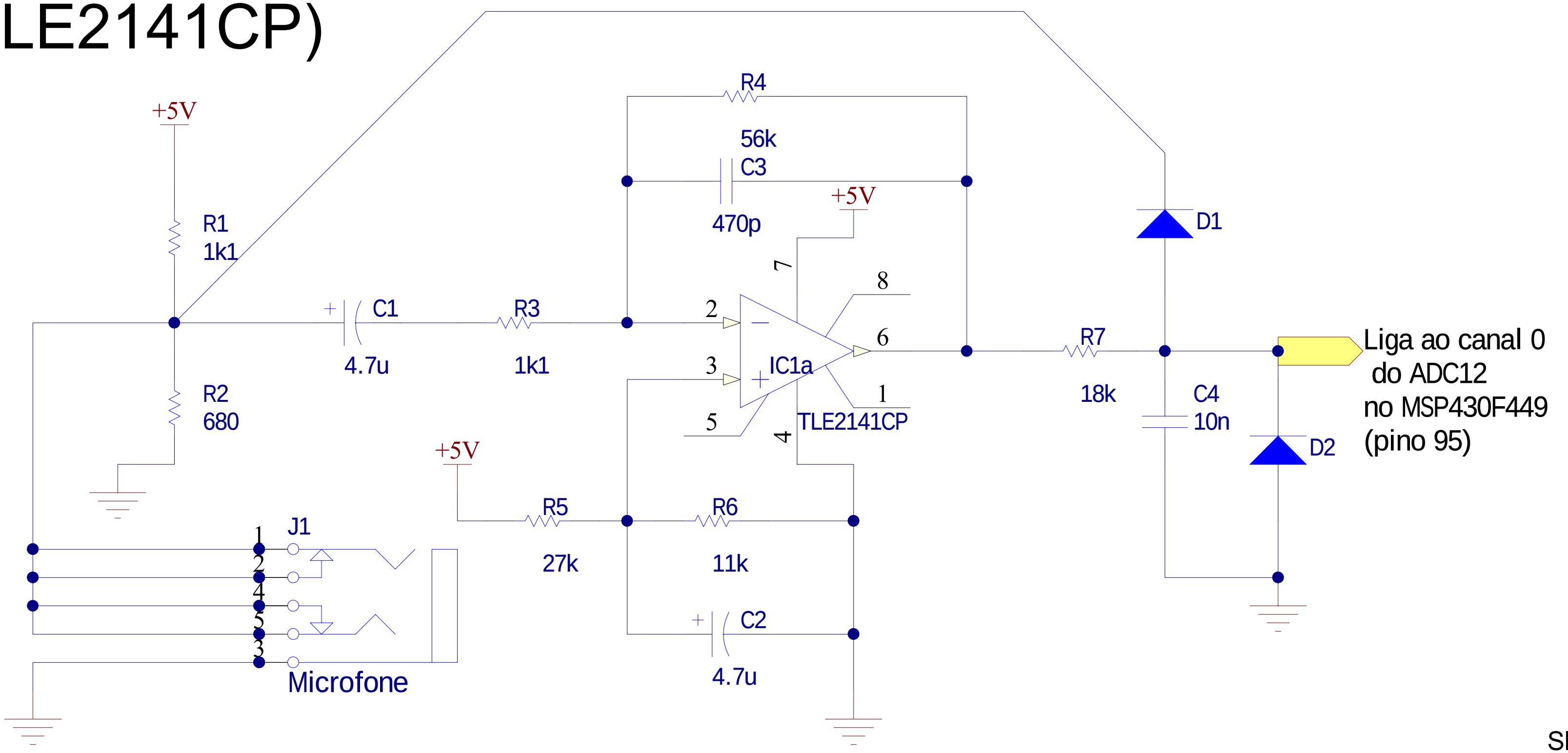
- 2 important states: Recording and Reproduction

- Recording state: 1st analog part and the digital one actives.
- During this time period, the data converted by the ADC12 will be stored in the flash memory.

- Reproduction state: digital part and the 2nd analog part will be actives.
- The voice signal is sent to the PWM output by TImer_B.
- A filter was developed to improve the PWM DAC signal, which is amplified and sent to the speaker.

Microphone and Conditioning system

- Pre-amplifying circuit and low pass filter
- Microphone: input jack 3,5 mm (J1)
- Amplifying stage: Active low pass filter (TI OA TLE2141CP)

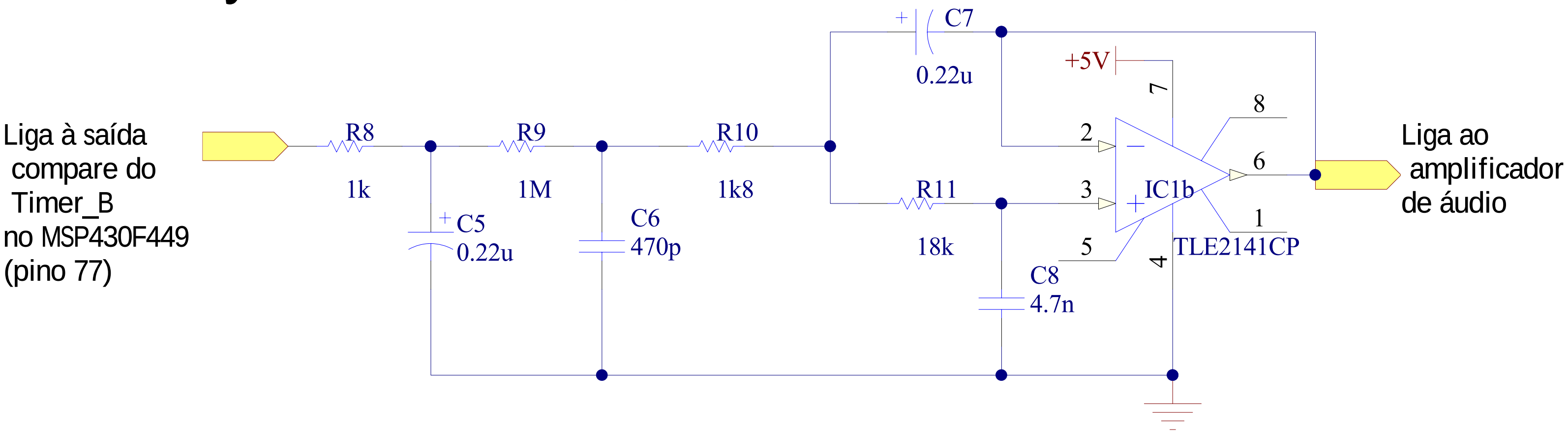


PWM DAC and active filter

- The PWM DAC is constituted by two RC series low pass filters.
- The digital signal in the TIMER_B compare output is converted to an analog signal.
- To reduce the attenuation amount of the signal produced by the filter, the cut off frequency of the filter should be near the band weight used at the ADC12 input.
- Cut off frequency: $f_{cut\ off}$ = 723 Hz

PWM DAC and active filter

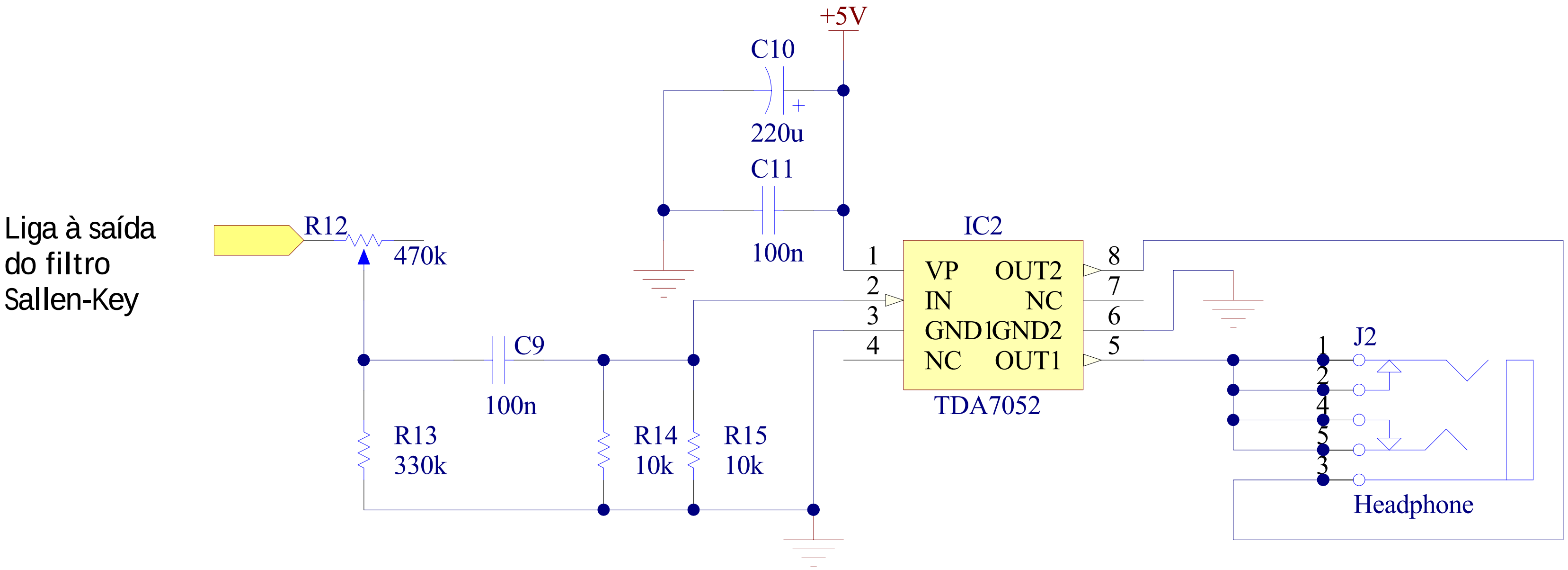
- Before amplifying the signal, it is used an Sallen-Key low pass filter with unitary gain.
- 2 resistors (R10 e R11),
- 2 condensers (C7 e C8)
- TI OA TLE2141CP with unitary gain (*buffer*)
  - Cut off frequency: $f_{cut\ off}$ = 870 Hz
  - Quality factor: Q = 1,97



Last Year Projects

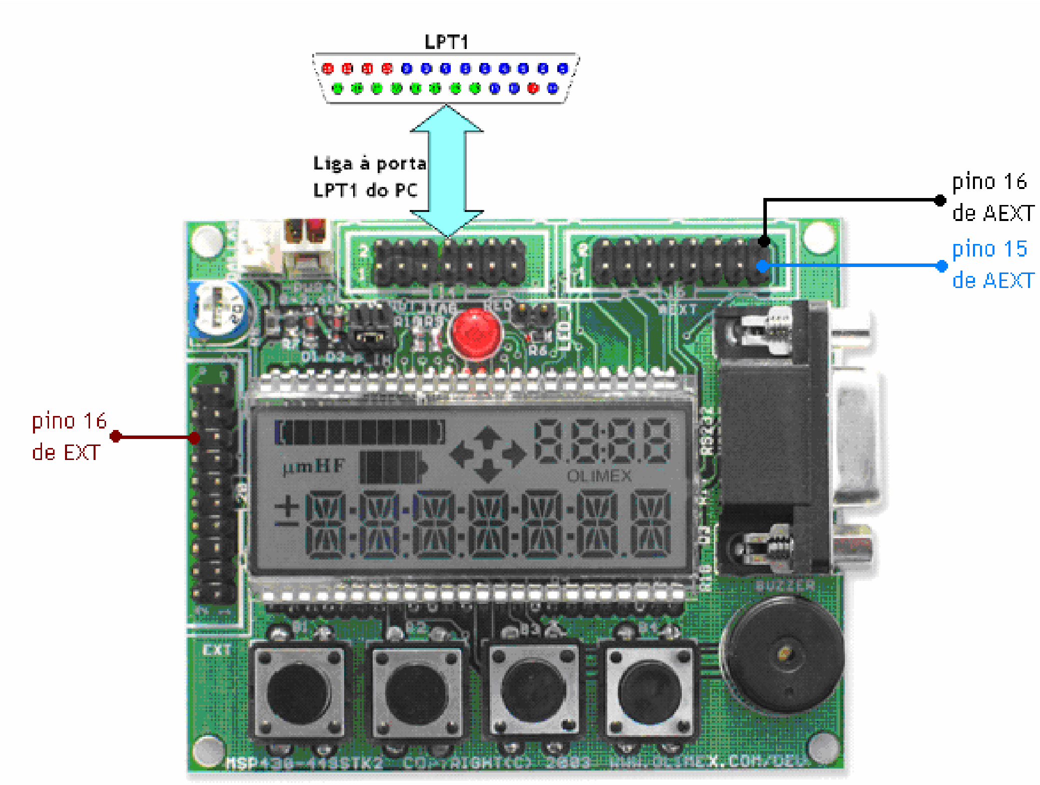
Audio amplifier

- Volume control: tension divider (R12 and R13)
- Limit the input audio signal level.
- Amplifying circuit: Audio amplifier Philips TDA7052



Using MSP430F449 OLIMEX kit

- 3 external interfaces: JTAG, AEXT and EXT
- AEXT interface - ADC12
- EXT interface – Access to multiplexed ports.



PWM signals - Software

- Erasing flash memory by segments

Step	Description
1	Stop Watchdog timer
2	Select clock source. f ≈ 444 kHz
3	Unblock flash memory
4	Set the erase mode by segments
5	Set to 0 the flash memory index which segment is to be erased
6	Unset the erase mode by segments
7	Block flash memory

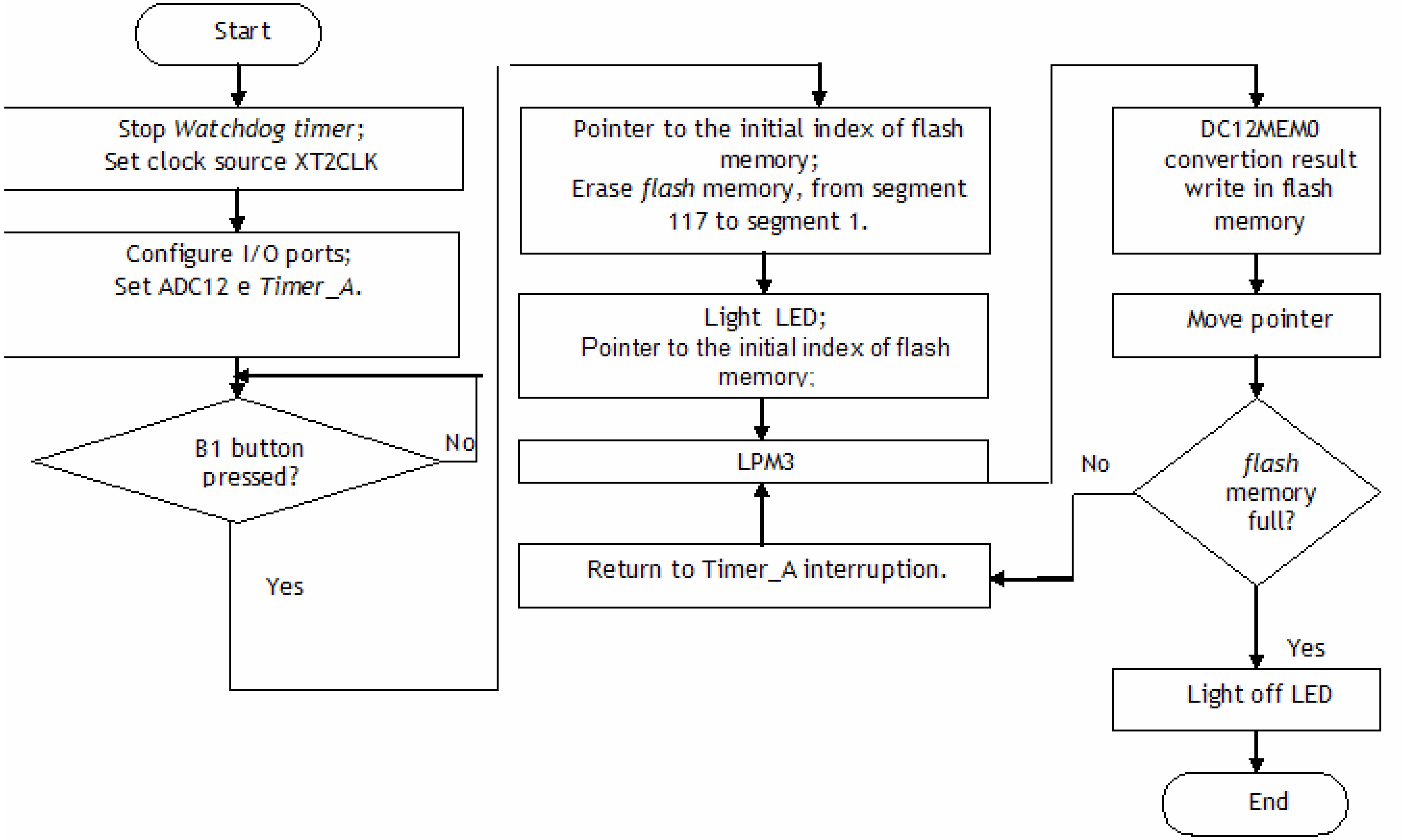
PWM signals - Software

- Write words in flash memory

Step	Description
1	Stop Watchdog timer
2	Unblock flash memory
3	Set the word writing mode
4	Attribute the memory index with a 16 bits data
5	Un set the word writing mode
6	Block flash memory

Storing voice signals - Software

- Flash memory used: 370 Kbytes
- Duration of speech: 15,3 sec



Using Timer_A - Software

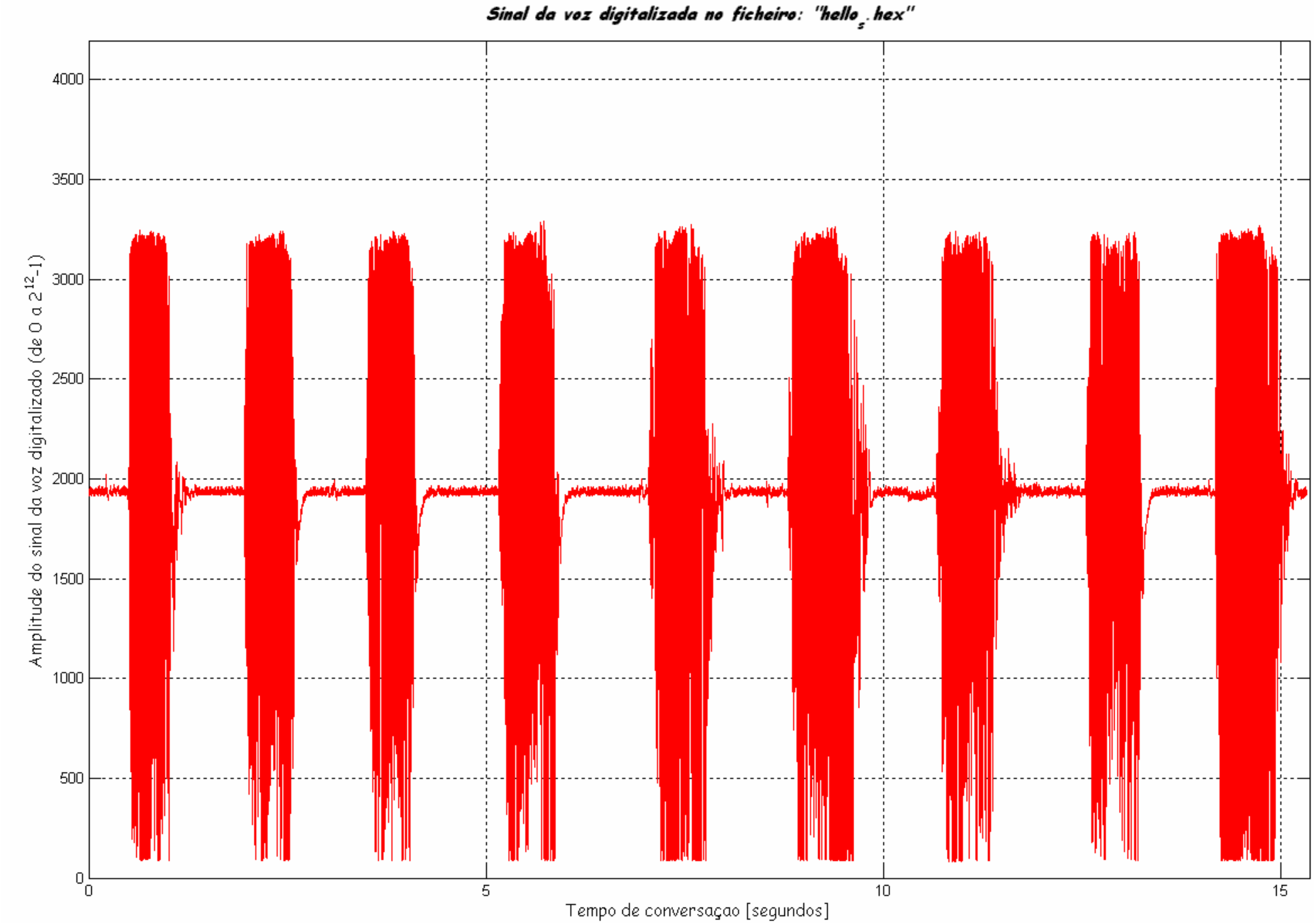
- Speech signal sampling frequency: 1953 Hz = 512 μsec.
- Generate an interruption to save the speech signal sample.
- Operation mode: up mode.
- Increase TAR until TAR = TACCR0 then generate interruption.

Using ADC12 - Software

- Conversion reference voltage: 0 – 3 V.
- Time of conversion: 5,8 μsec.

MatLab verification

- Word saved in flash memory: “hello”



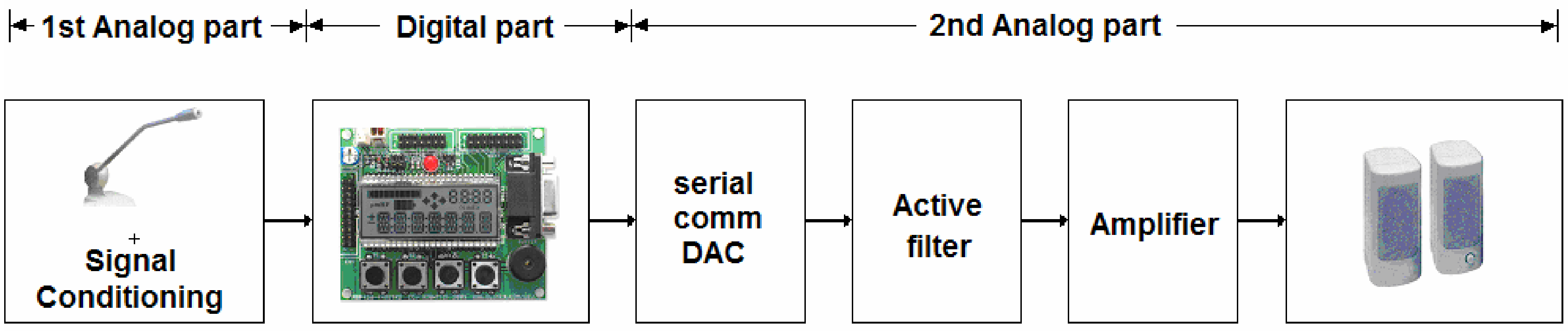
Using Timer_B - Software

- PWM outputs each 512 µsec.
- Each signal sample stored in flash memory is sent to the PWM DAC through Timer_B interruption.
- Operation mode: reset/set.

Serial communication DAC

- 2nd choice for the global architecture of the system.
- Substitution of the PWM DAC for a serial communication DAC.

- Using the same peripherals but also the USART in SPI mode.
- Allows sending the serial data synchronized to the 12 bits DAC (TI TLV5616CP).

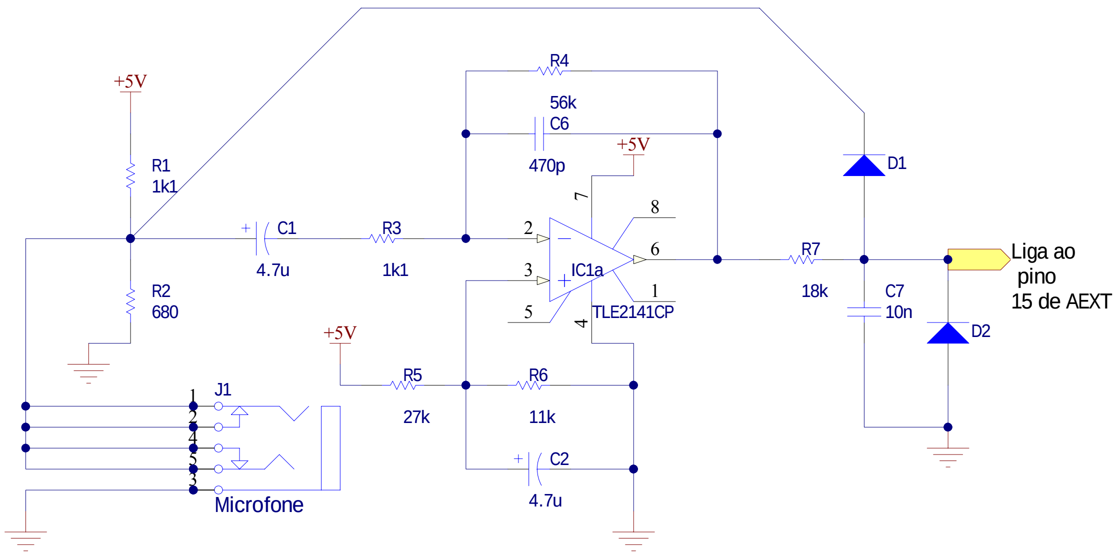


Using USART - Software

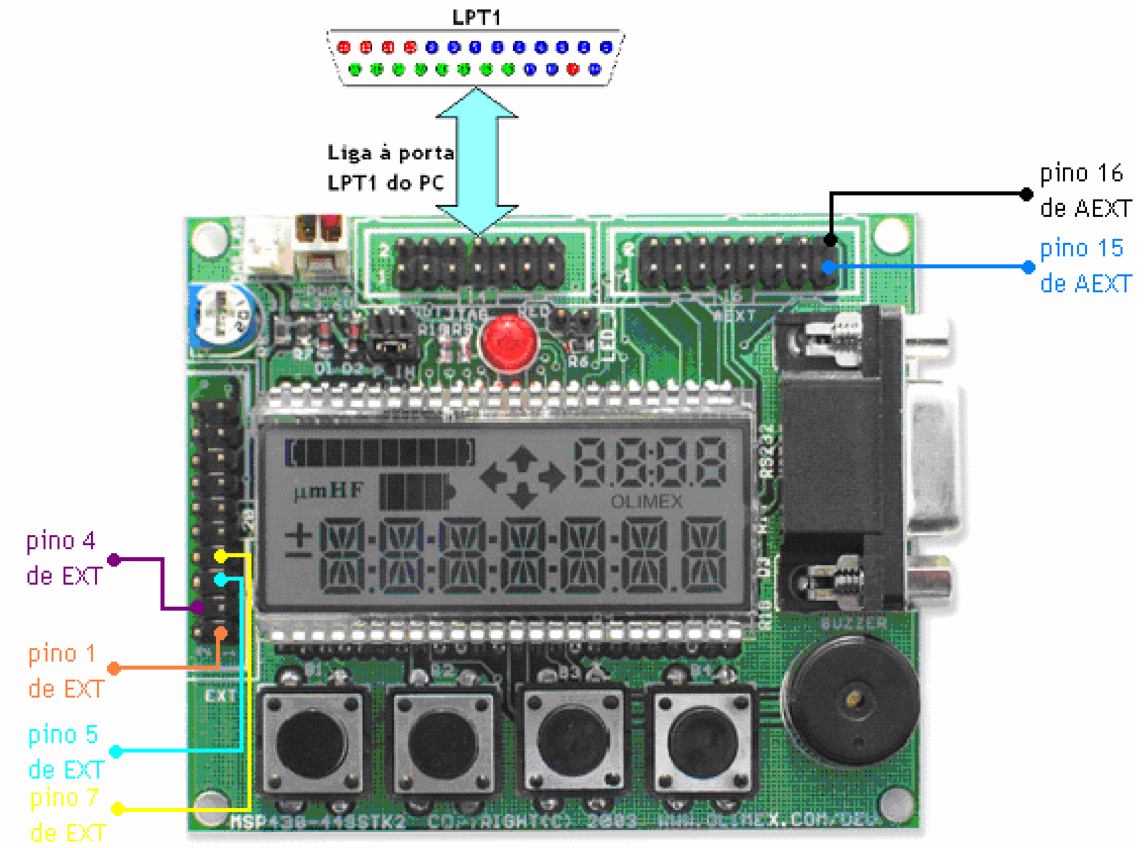
- 2nd choice for the global architecture of the system.
- Substitution of the PWM DAC for a serial communication DAC.
- Using the same peripherals but also the USART in SPI mode.
- Allows sending the serial data synchronized to the 12 bits DAC (TI TLV5616CP).
- In synchronous mode, the USART allows to connect the MSP430 to an external device through: SIMO, SOMI, UCLK and STE.
- For correct operation of the USART in SPI mode, the bit SYNC in the U*x*CTL control register must be active.

Serial communication DAC

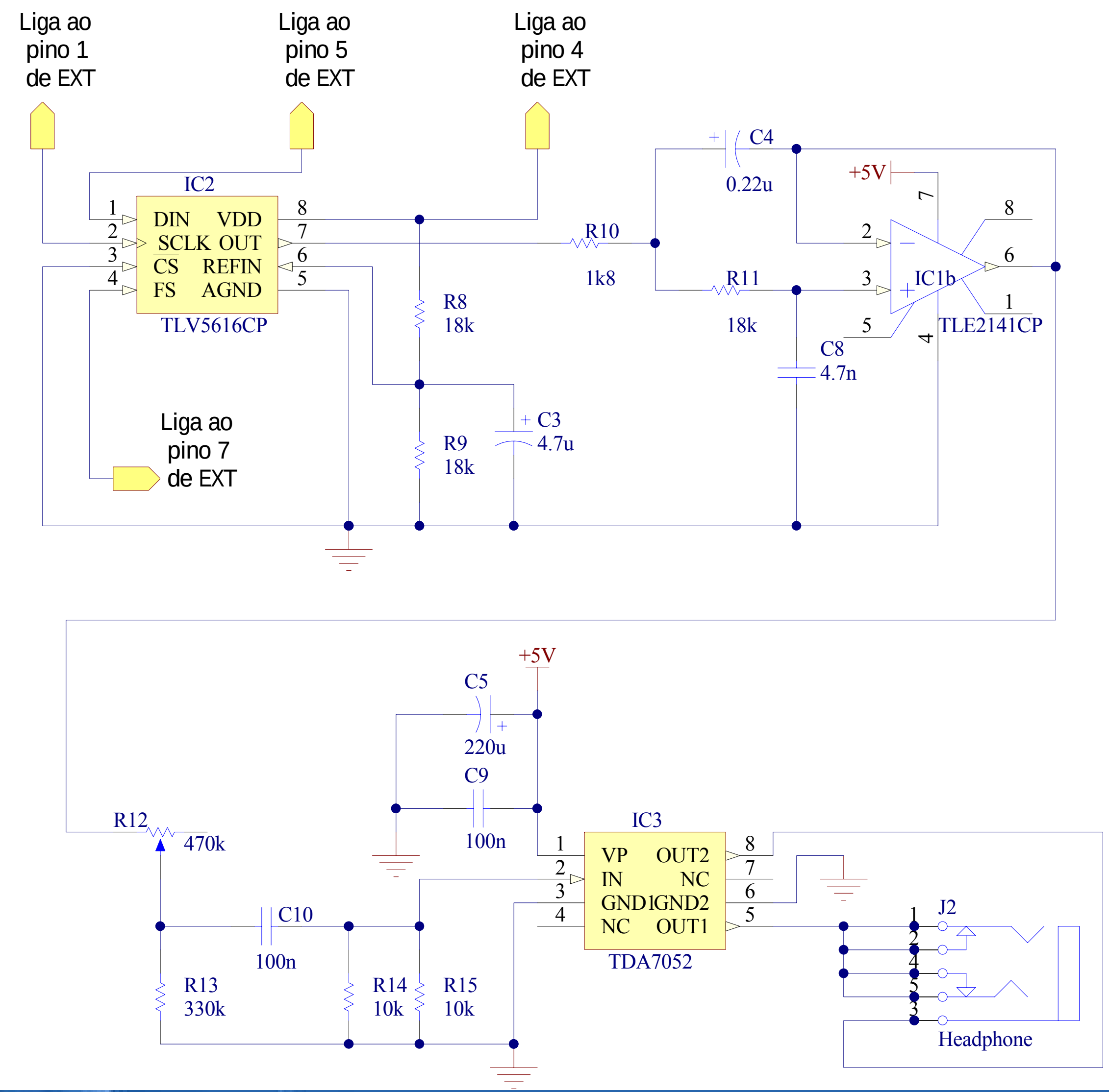
- Using the Olimex kit MSP430F449, the connections are:



Serial communication DAC



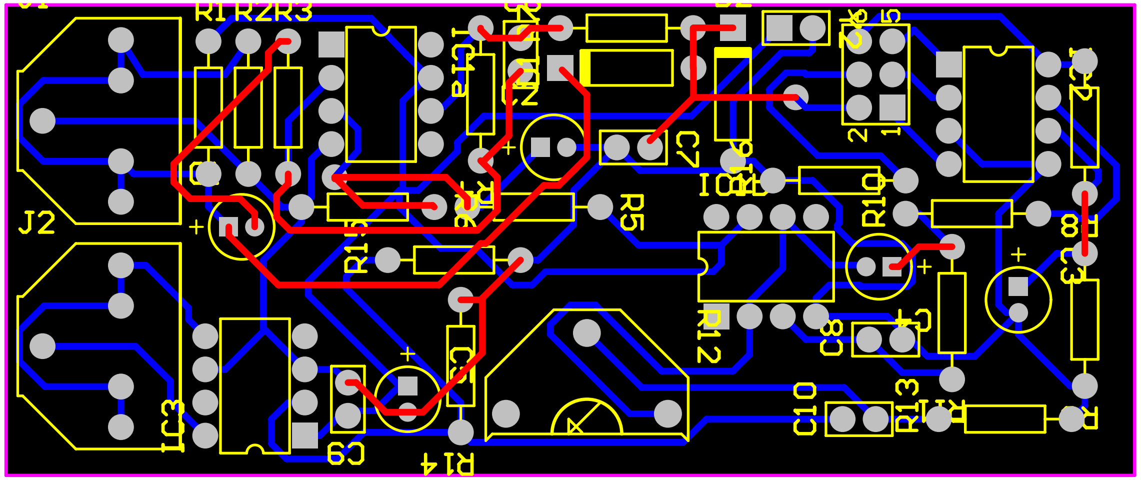
Serial communication DAC

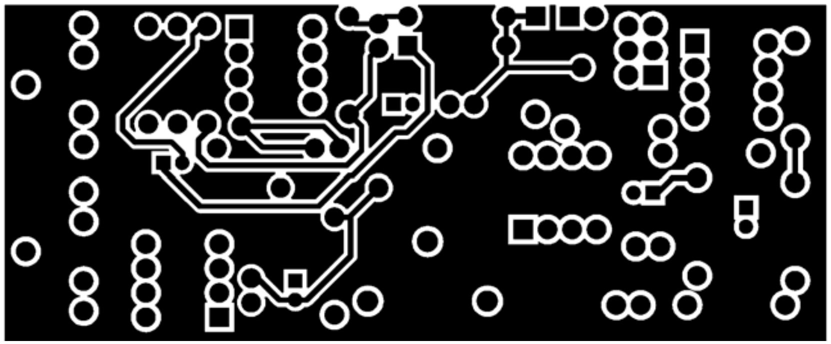


Last Year Projects

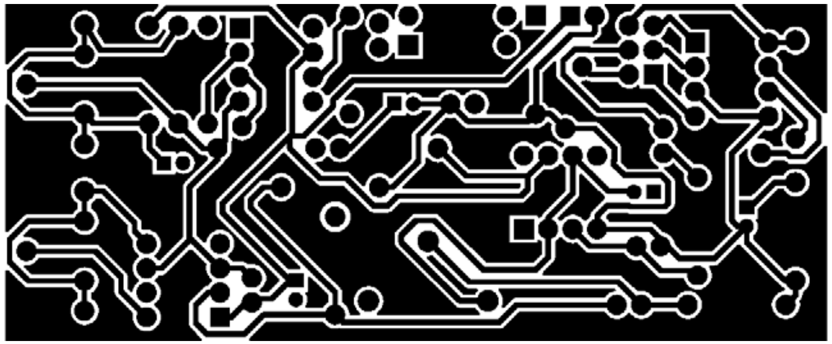
Last Year Projects

Breadboard development



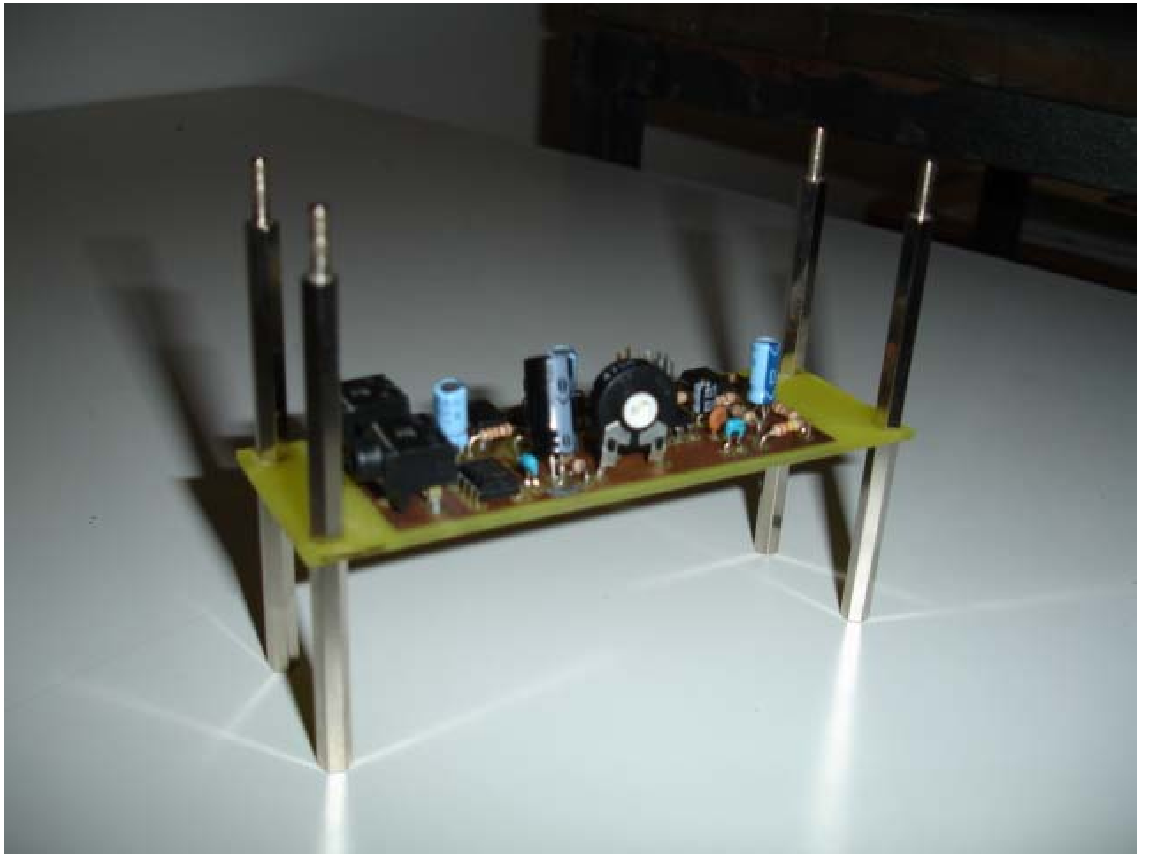


Upper connections



Lower connections

Breadboard development



Project conclusions

- Using PWM signals:
  - Low quality of the sound reproduced
  - Due to the sampling frequency (1953.125 Hz)
  - Speech duration: 15,3 sec

- Using serial communication DAC TLV5616CP
  - Higher quality of the sound reproduced
  - Speech duration: 3 sec

- To include this sound recording and reproduction system in the RoboSapien, it is necessary to include an external memory in MSP430 to increase the duration of the speech.

Project conclusions

- The voice commands recognition by the RoboSapien needs:
  - Voice acquisition;
  - Parameters extraction;
  - Pattern recognition.

- Voice acquisition: ADC12.
- Parameterization algorithm: development of a software code to extract the characteristics of the voice acquired, in a manner of extraction commands interpretation through their pattern.
- Pattern recognition: through successive comparisons with a reference pattern.

Summary

- Presentation of the teaching structure at UBI in classes that use microprocessors;
- Exposition of the items covered in the substitution of the RoboSapien microcontroller by TI MSP430, untitled “RoboSapien powered by MSP430”;
- Presentation of last year degree projects using MSP430 as the “sound recording and reproduction system for the Robosapien”.

Acknowledgments

- This project was partially funded by Texas Instruments;

- The authors would like to particularly acknowledge the help provided by:

- Christian Speck

  (*TI Europe Business Development*);

- Robert Owen

  (*TI University Programme Manager: Education & Communications*).