DIPARTIMENTO di INGEGNERIA dell'INFORMAZIONE
SUN Ingegneria
SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI

Real Time Control of an Anthropomorphic Robotic Arm using FPGA

Advisor:
Prof. Ciro Natale

Students:
Francesco Castaldo
Andrea Cirillo
Pasquale Cirillo
Umberto Ferrara
Luigi Palmieri

Objective

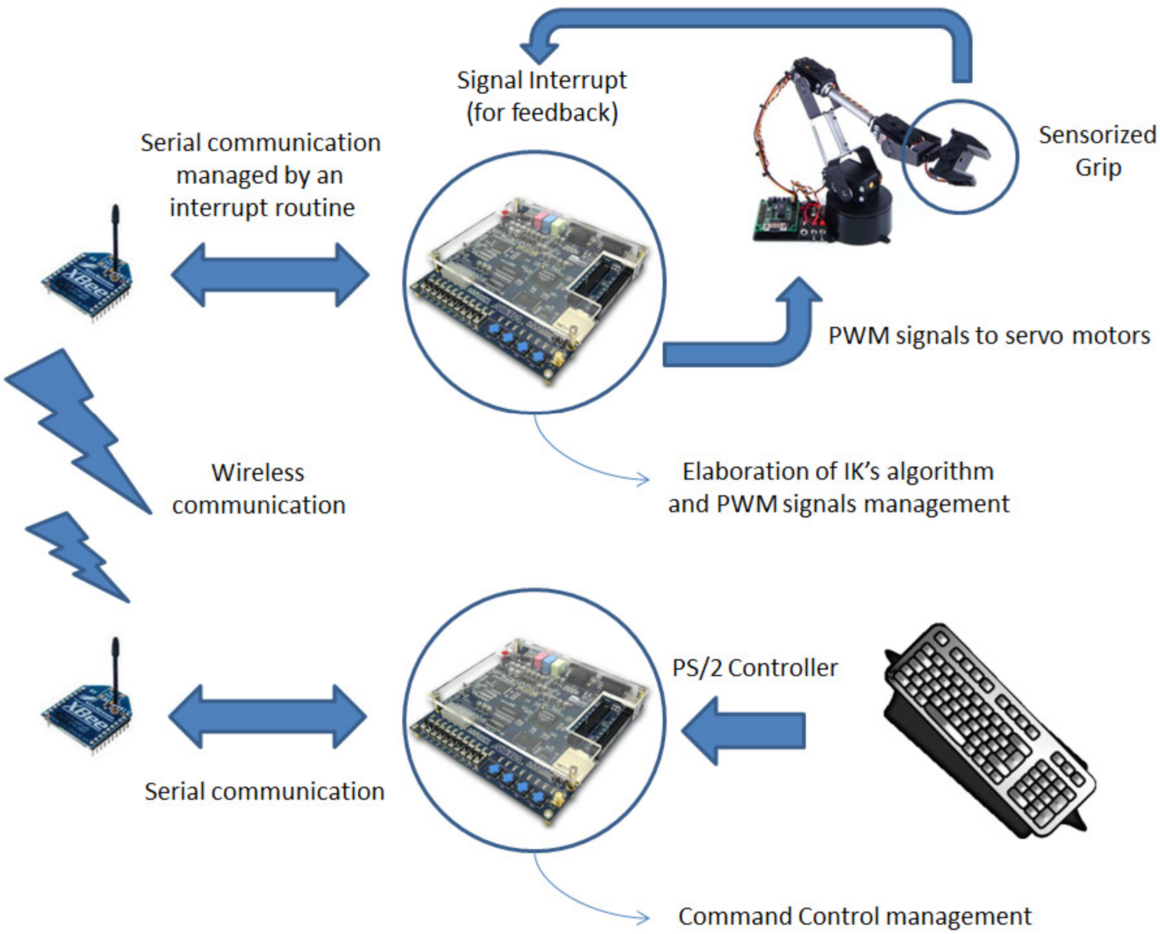
Introduction

- The project consists to make an anthropomorphic robotic arm controlled in real-time by user with a wireless controller.
- The whole system is FPGA-based and it doesn't use a personal computer.

Some applications

- The idea is to realize a low cost control system that can be used in some critical applications:
  - Rescue missions;
  - Remote manipulation.

Architecture

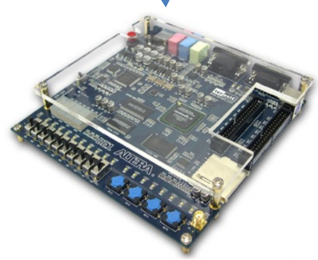


- Two Altera DE1 Boards;
- One PS/2 Keyboard;
- Two Xbee Module;
- An anthropomorphic robotic arm with spherical wrist (6 DOF);
- An home-made optoelectronic force-sensor.

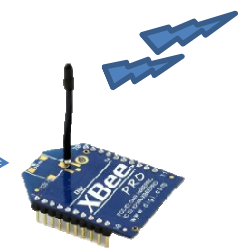
Functioning (1/2)

User sends a remote command pressing a button of the keyboard.

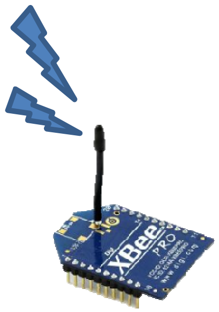
FPGA captures the scancode from PS/2 interface and it sends the command byte to the Xbee module.



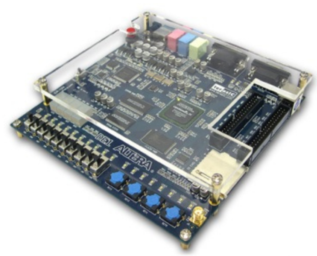
The transmitter Xbee module sends the information to the receiver module.



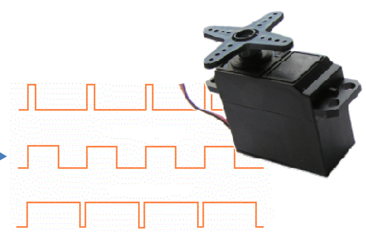
Functioning (2/2)



The Xbee Module sends the received command to FPGA control unit.



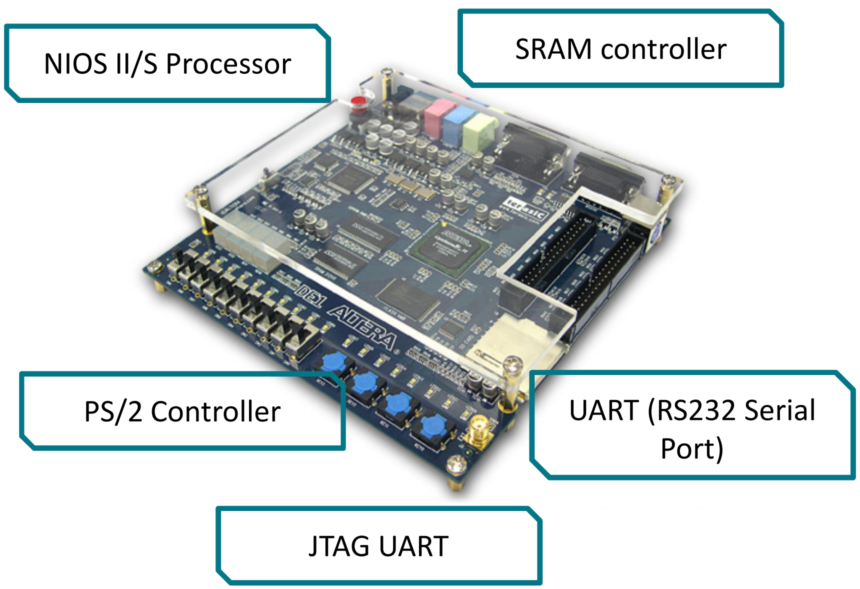
FPGA elaborates information, changes the wirst position, resolves the inverse kinematic algorithm.



FPGA updates the PWM signals for the seven servo motors.

The arm moves in real-time and can receive a feedback from the sensor mounted on the end-effector.

Soft-Core NIOS II (1/2)



FPGA for User Interface

Use	Conn...	Name	Description
☑		⊟ cpu	Nios II Processor
		instruction_master	Avalon Memory Mapped Master
		data_master	Avalon Memory Mapped Master
		jtag_debug_module	Avalon Memory Mapped Slave
☑		⊟ sram	SRAM/SSRAM Controller
		avalon_sram_slave	Avalon Memory Mapped Slave
☑		⊟ ps2	PS2 Controller
		avalon_ps2_slave	Avalon Memory Mapped Slave
☑		⊟ jtag_uart	JTAG UART
		avalon_jtag_slave	Avalon Memory Mapped Slave
☑		⊟ uart0	UART (RS-232 Serial Port)
		s1	Avalon Memory Mapped Slave

Compilation Report

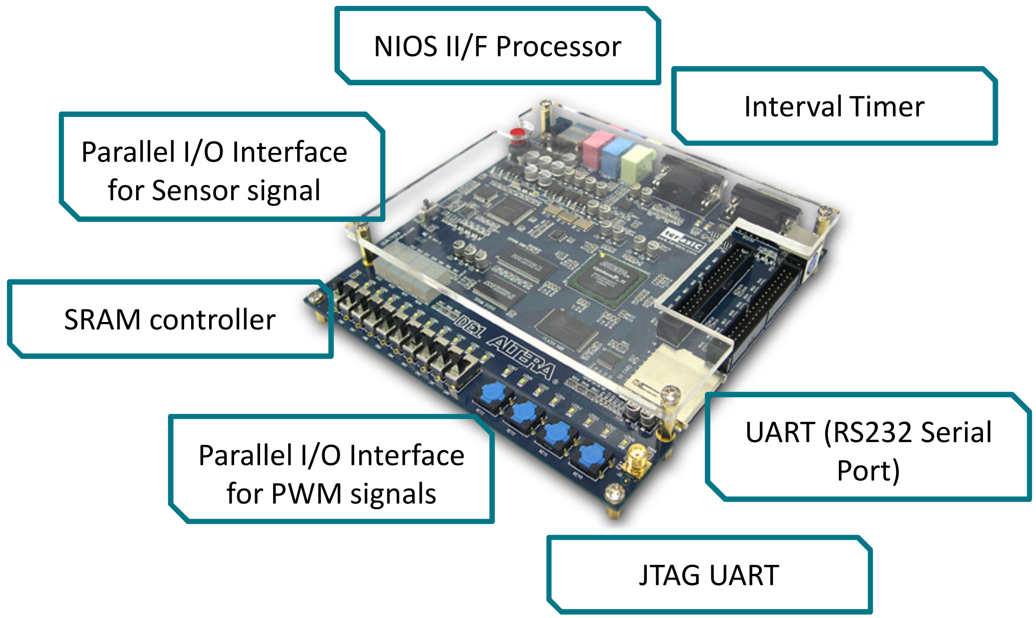
Flow Summary	
Flow Status	Successful - Tue Nov 22 00:04:47 2011
Quartus II Version	11.0 Build 157 04/27/2011 SJ Web Edition
Revision Name	key
Top-level Entity Name	key
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	2,855 / 18,752 (15 %)
Total combinational functions	2,664 / 18,752 (14 %)
Dedicated logic registers	1,722 / 18,752 (9 %)
Total registers	1764
Total pins	45 / 315 (14 %)
Total virtual pins	0
Total memory bits	48,256 / 239,616 (20 %)
Embedded Multiplier 9-bit elements	4 / 52 (8 %)
Total PLLs	0 / 4 (0 %)

Soft-Core NIOS II (2/2)

Use	Conn...	Name	Description
☑		⊟ cpu	Nios II Processor
		instruction_master	Avalon Memory Mapped Master
		data_master	Avalon Memory Mapped Master
		jtag_debug_module	Avalon Memory Mapped Slave
☑		⊟ timer	Interval Timer
		s1	Avalon Memory Mapped Slave
☑		⊟ pwm	PIO (Parallel I/O)
		s1	Avalon Memory Mapped Slave
☑		⊟ jtag_uart	JTAG UART
		avalon_jtag_slave	Avalon Memory Mapped Slave
☑		⊟ uart	UART (RS-232 Serial Port)
		s1	Avalon Memory Mapped Slave
☑		⊟ sram	SRAM/SSRAM Controller
		avalon_sram_slave	Avalon Memory Mapped Slave
☑		⊟ grasp	PIO (Parallel I/O)
		s1	Avalon Memory Mapped Slave
☑		⊟ sensor	PIO (Parallel I/O)
		s1	Avalon Memory Mapped Slave

Compilation Report

Flow Summary	
Flow Status	Successful - Sun Sep 18 10:33:49 2011
Quartus II Version	11.0 Build 157 04/27/2011 SJ Web Edition
Revision Name	roboticArm
Top-level Entity Name	roboticArm
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	3,437 / 18,752 (18 %)
Total combinational functions	3,072 / 18,752 (16 %)
Dedicated logic registers	2,168 / 18,752 (12 %)
Total registers	2168
Total pins	51 / 315 (16 %)
Total virtual pins	0
Total memory bits	64,704 / 239,616 (27 %)
Embedded Multiplier 9-bit elements	4 / 52 (8 %)
Total PLLs	0 / 4 (0 %)



FPGA for Control Unit

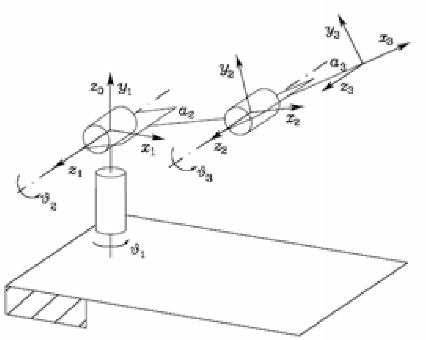
Inverse Kinematic (1/2)

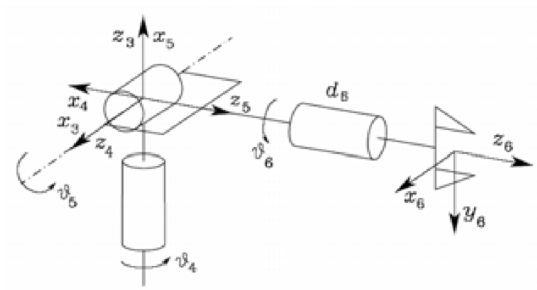
The inverse kinematic problem is difficult to solve:

- Non-linear equations (sine, cosine in rotation matrices);
- The existence of multiple solutions;
- The possible non-existence of a solution;
- Singularities.

IK Simplifications:

- Decouple the problem into independent subproblems:
  - determining the inverse solution to the problem of positioning;
  - determining the inverse solution to the problem of orientation.



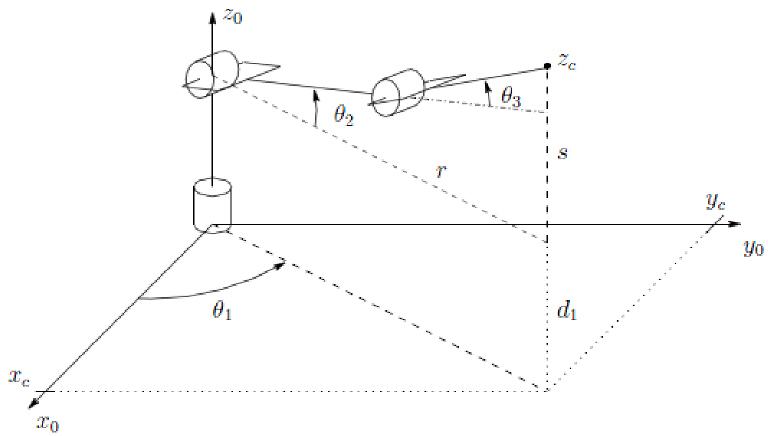


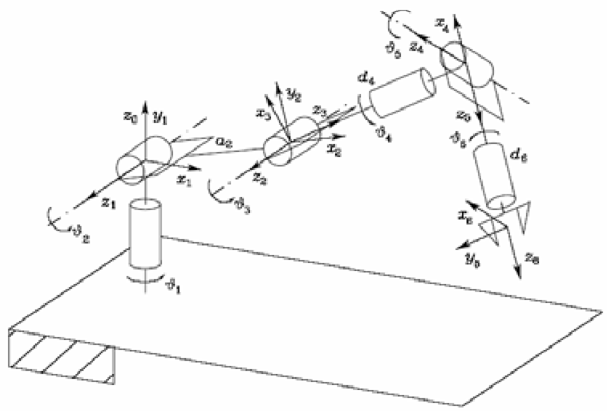
Inverse Kinematic (2/2)

Our Implementation

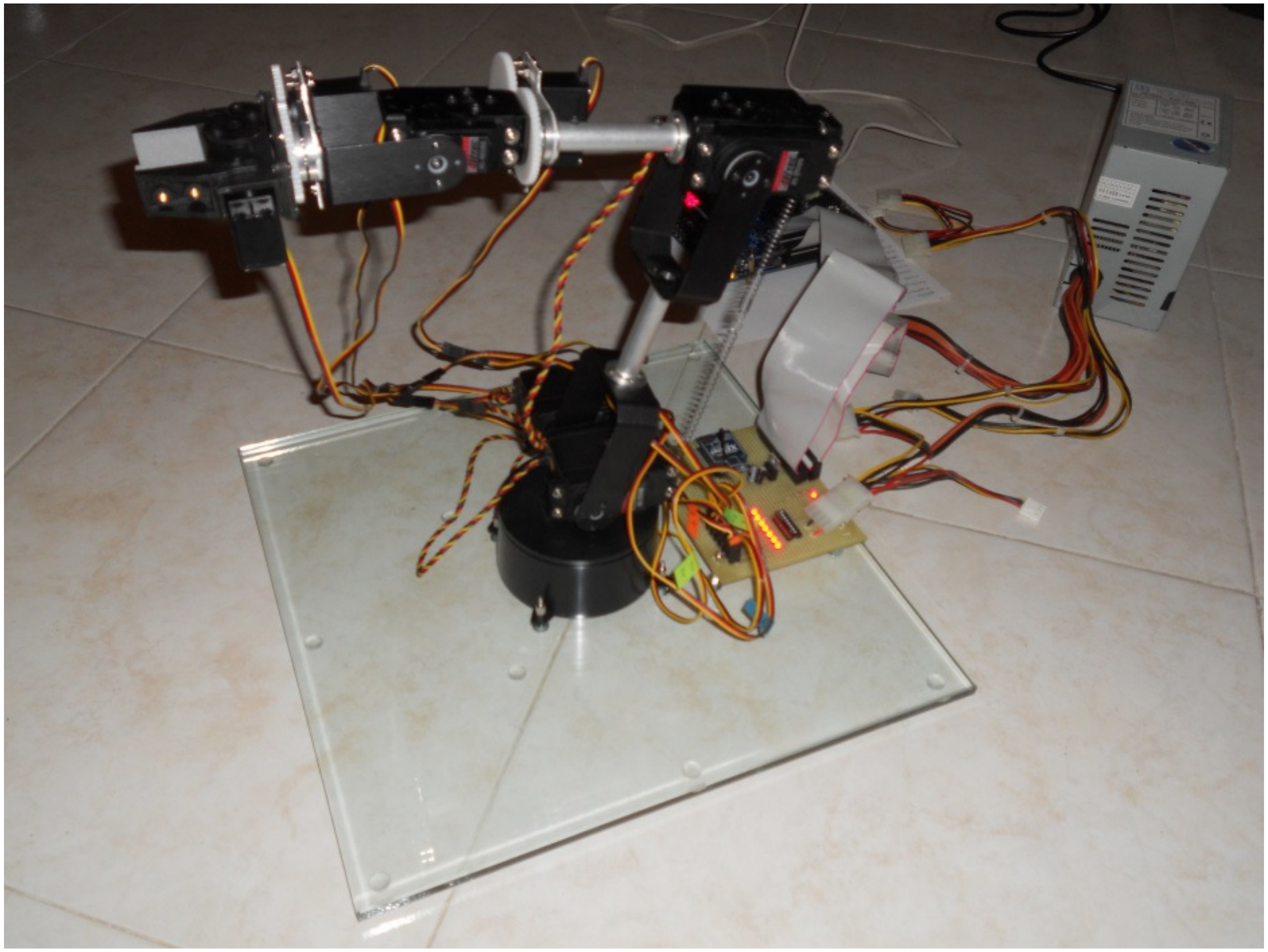
In the system, the user controls the position of the wirst:

- The IK algorithm calculates the 3 first degree of the arm ($\vartheta_1, \vartheta_2, \vartheta_3$);
- The user sets the orientation of the end-effector ($\vartheta_4, \vartheta_5, \vartheta_6$).



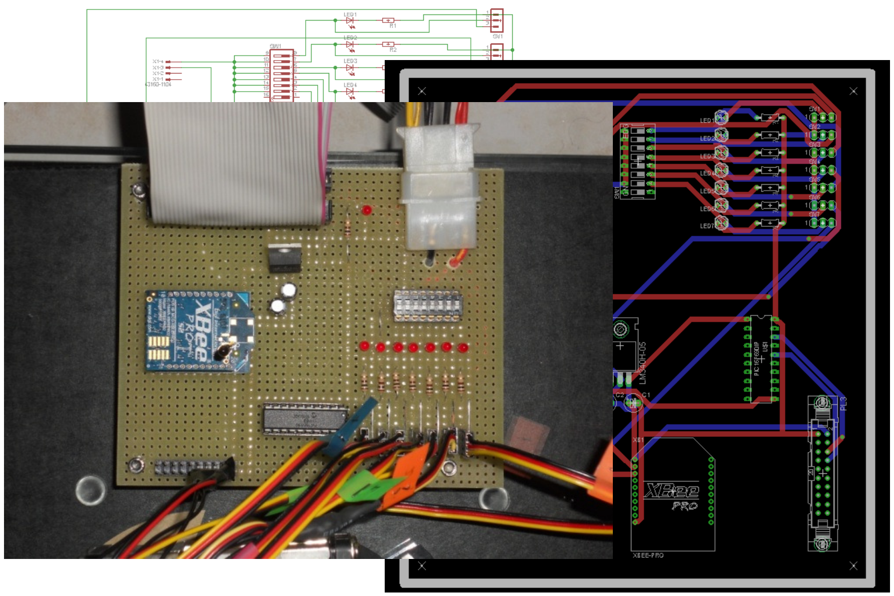


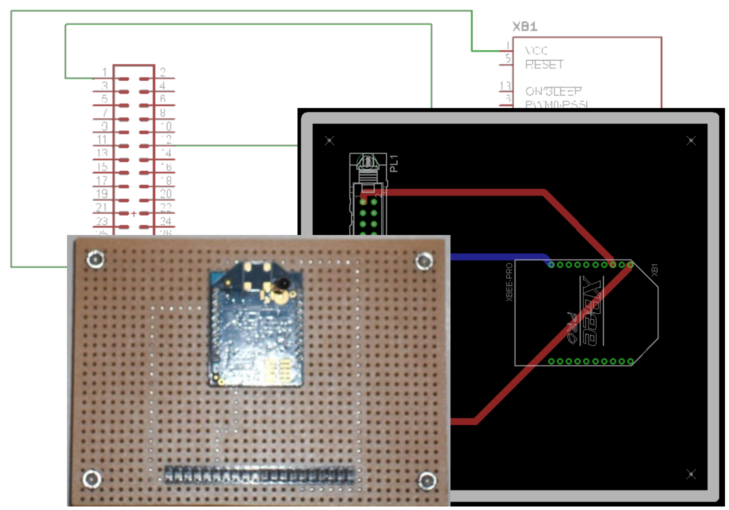
The Arm



Interfacing FPGAs

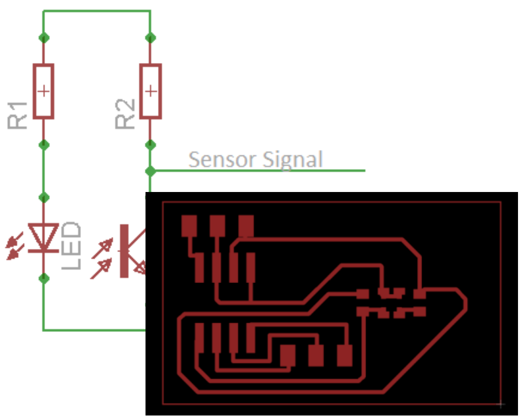
- We use the expansion headers of the DE1 Board development kit (GPIO_0 and GPIO_1) to interface the FPGAs with the XBEE modules and with the arm.
- Two boards have been made:
    - One for the manipulator and the FPGA that handles the control signals for servo motors;
    - Another one for the FPGA on which the controller is implemented.

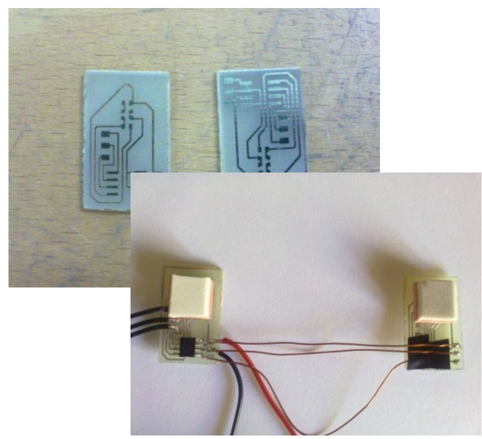


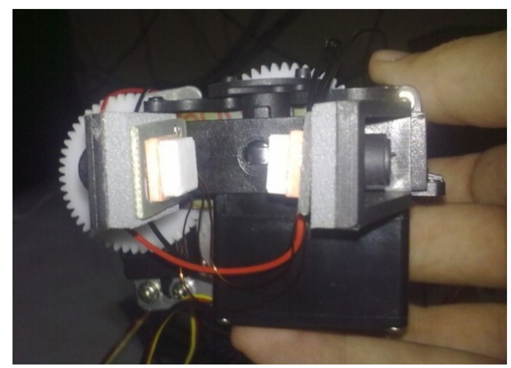


Force Sensor

- The sensor developed for the gripper provides information about the successful operation of grasping:
  - It estimates the contact force;
  - Simply, comparing the voltage value with a predetermined threshold voltage, it gives information about the contact between two bodies.







Implementation (1/4)

Principal problems:

- Commands acquisition;
- Interfacing with Xbee Module;
- Implementation of Inverse Kinematic Algorithm;
- PWM Signals generation;
- Management sensor feedback.

Commands acquisition:

- Use PS/2 Controller;
- Decode the keyboard scancode received;

```
alt_up_ps2_dev* alt_up_ps2_open_dev(const char *name)
void alt_up_ps2_init(alt_up_ps2_dev *ps2)
int decode_scancode(alt_up_ps2_dev *ps2, KB_CODE_TYPE *decode_mode, alt_u8 *buf, char *ascii)
```

Implementation (2/4)

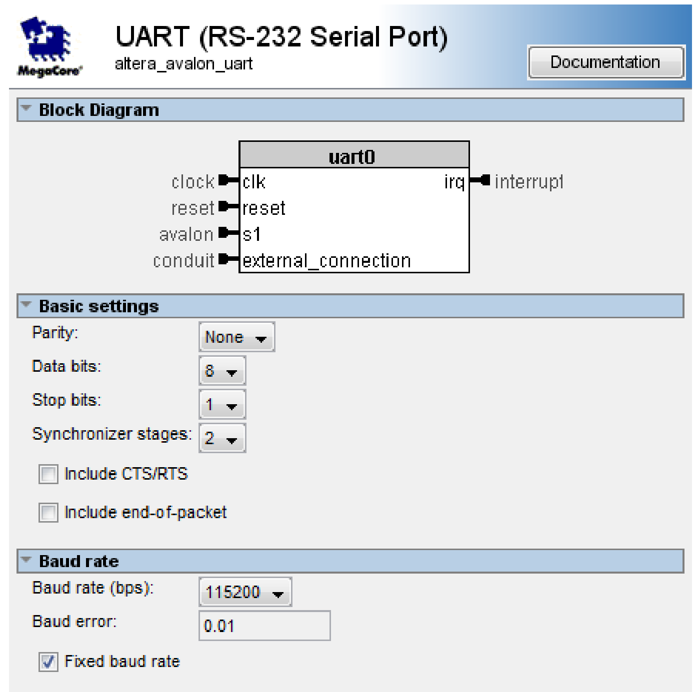
Interfacing with Xbee Module:

- Use UART Interface:
  - BaudRate: 115200bps;
  - Parity: NONE;
  - DATA Bits: 8
  - Stop Bits: 1

```
IOWR_ALTERA_AVALON_UART_TXDATA(base, data)
```

- Receive with UART Interrupt:

```
alt_irq_register(UART_IRQ, 0, uart_ISR);
…
command = IORD_ALTERA_AVALON_UART_RXDATA(UART_BASE);
```



Implementation (3/4)

Implementation of Inverse Kinematic Algorithm:

- Include math.h library for atan2() function, non linear sine and cosine function;
- Implementation of matrix transpost function;
- Implementation of matrix product function.

PWM Signals generation:

- Use Timer to generate interrupt;
- Use GPIO pins;
- Signal frequency: 50 Hz;
- Update Duty Cicle after IK algorithm execution.

```
alt_irq_register(TIMER_IRQ, 0, timer_ISR);
…
IOWR_ALTERA_AVALON_TIMER_STATUS(TIMER_BASE, 0);
```

Implementation (4/4)

Management sensor feedback:

- Use GPIO Interrupt;
- Stop the motor of the grip when interrupt occurs.

```
alt_irq_register(SENSOR_IRQ, 0, sensor_ISR);
```

DIPARTIMENTO di INGEGNERIA dell'INFORMAZIONE
SUN Ingegneria
SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI

Future Developments

- Add angular sensors to control arm in feedback to improve the precision of the movements.
- Add camera on the grip to view the target position in the workspace.
- Give to the arm the possibility to move as a mobile robot.
- Replace the keyboard with a R/C controller to improve the movement flexibility.

The manipulator at work…

Video