Control of robotic arm and gripper

Cristian Enrico Capalbo

December 3, 2018

1 Introduction

In the original design of the robot, the degrees of freedom of the arm and the gripper (the end effector) are controlled by means of a remote controller.

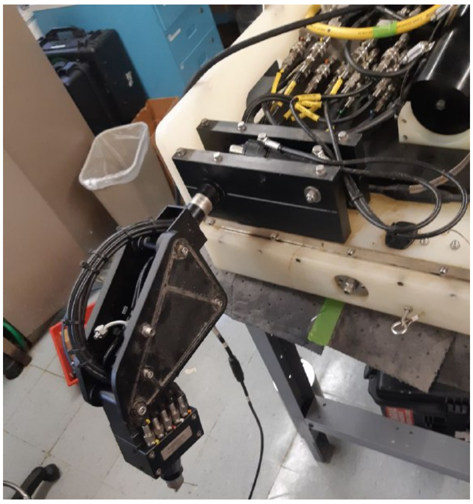


Figure 1: Robotic arm

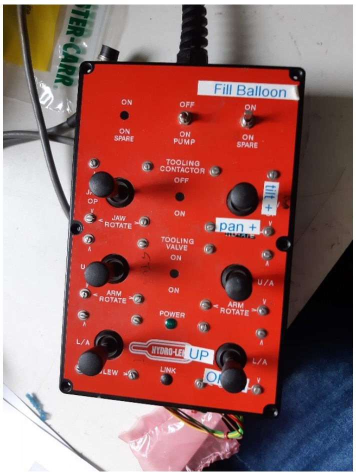


Figure 2: Remote controller

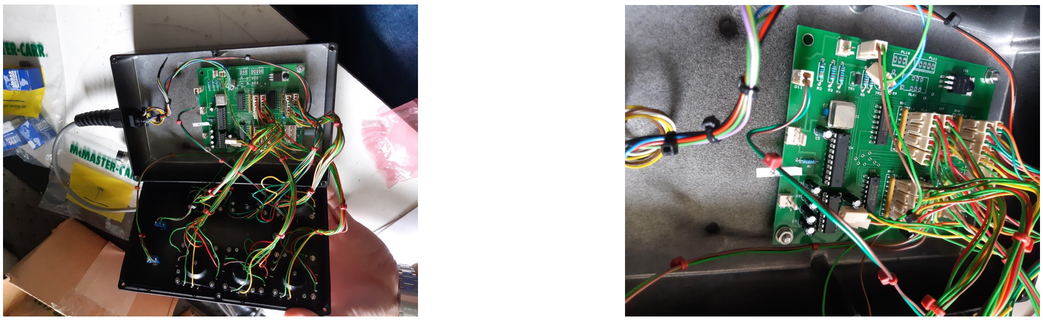
Although this is the best solution, as it was at the start of the experience it showed a sort of a lag, since there was a delay between the moment in which the command was sent and the effective movement; also, if the command was too short the arm didn't move at all.
At the starting point the reason of this problem was unknown, it could either be the control electronics or the hydraulic actuation of the system. We choose to examine first the electronics, then go on with the hydraulic valves if necessary.

2 Electronics

The electronic system consists initially of a printed board, housed in the remote, to which all the cables from the several joysticks are connected. This board reads the analogic signals and sends them to the on-board controller of the valves through an ethernet connection. The printed board contains some chips that could cause the delay, and each of them has been analysed.

2.1 Multiplexer

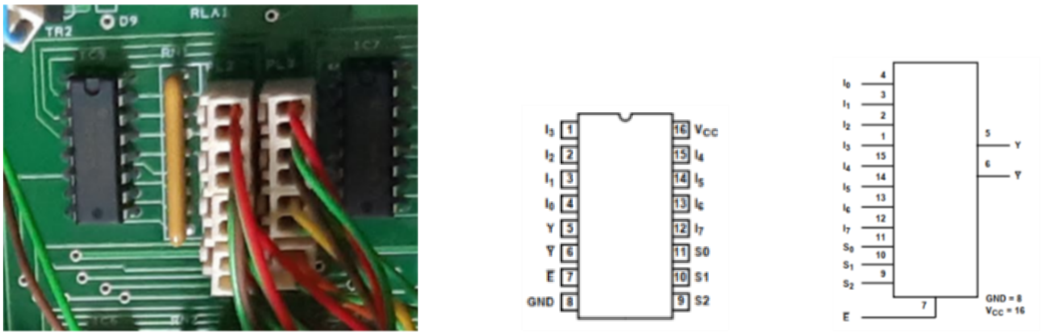
The first element analysed is a Multiplexer (TI CD74HC151E); in the board we can see three of them, each connected to the outputs of two joysticks. The function of the multiplexer is to read the 8 signals in input and giving as an output a single signal. To check if the delay was introduced in the system by this component we didn't need to know exactly how it works; what we did is simply connecting the output port of the multiplexer to an oscilloscope and verify that the change in the output happened instantly after the movement of the joystick. Being this verified we passed to the next element.



(a) Internal parts of the remote

(b) Electrical board of the remote

Figure 3: Components of the remote



(a) Multiplexer (TI CD74HC151E)

(b) Multiplexer ports configuration

(c) Multiplexer functioning scheme

Figure 4: Multiplexer

2.2 Microcontroller

The second element analysed, connected right after the three multiplexers, is an 8-bit microcontroller (ATMEL AT89C2051). As in the precedent case, we didn't need to know the actual functioning logic behind this component; all we needed to know is what was the output. From the connection to the other elements of the board we understood that it gave as output a serial signal in TTL/COMS format. To inspect this signal the output port was connected to an Mbed platform in turn connected to a computer through an USB port. A simple code was loaded in the platform that permitted to read the serial data from the microcontroller output and sent it to the computer; on this last one a software read and printed the messages. In this way we could see that the microcontroller maps every configuration of the joystick in a single Comma Separated Values string whit 5 numbers of 2 digits each. We could also see that there's no delay in this signal.

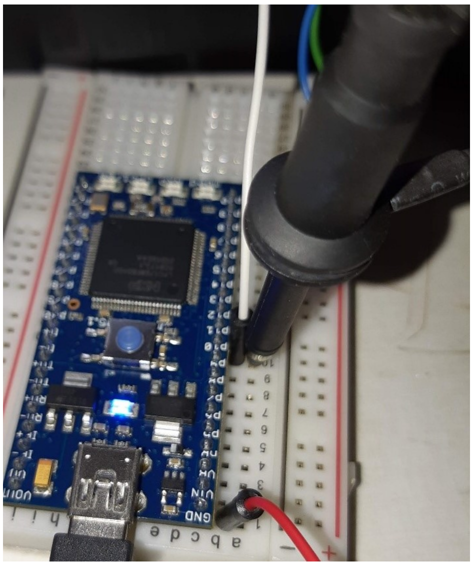


Figure 5: mbed configuration

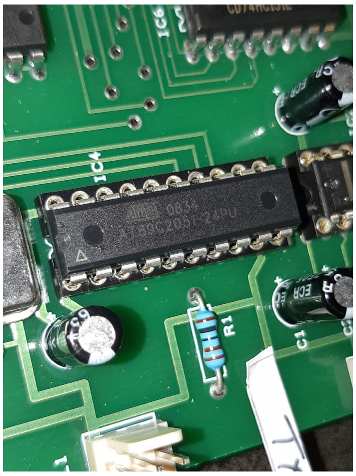
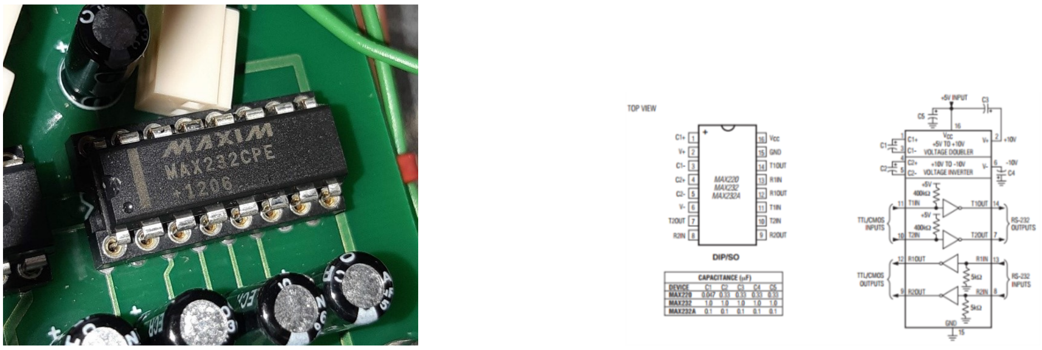


Figure 6: microcontroller (ATMEL AT89C2051)

2.3 RS232 driver

The last element in the remote electronics is a driver (MAX232) that has the only function of a communication interface between TTL/CMOS and RS232 standard that is the one used as output to communicate with the electronics on the ROV. Since it has multiple channels to actuate the translation in both ways, we took the first output (RS232) and translated it again in TTL/CMOS in order to read it with the same Mbed system. Also in this case we couldn't see any delay, so the next step was to examine the onboard electronics.



CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

(a) MAXIM MAX232CPE

(b) max232 functioning scheme

Figure 7: MAX232