User Documentation :

Controlling a 5R-robot by using a BBB controller

Apollo 20 team : Loïc Kerboriou, Justin Bezieau, Nicolas Cheron, Johann Huber
Polytech Sorbonne
January 2019

Summary:

Introduction

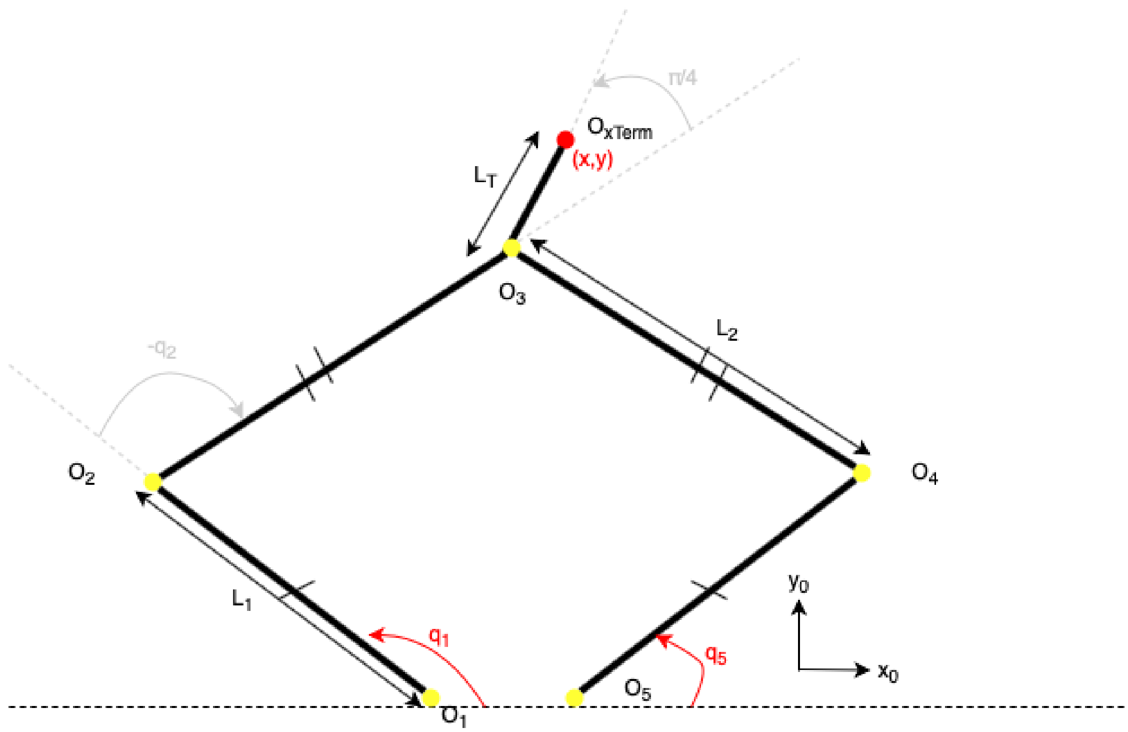
This document explains how to use the beaglebone black controller in order to draw forms with a 5r-robot. The purpose of the project is to reach a drawing error under the millimeter. The Apollo 20 project achieved the following goals :

- building a 5r plan robot which can follow a continuous trajectory under 3mm error
- implementing both direct and inverse geometric robots
- designing a Graphical User Interface to control the robot

Used languages : C++ (beaglebone black), python (on the workstation : GUI)

Required libraries : PyQt5, errno, subprocess, socket, sys, threading, time

Kinematics



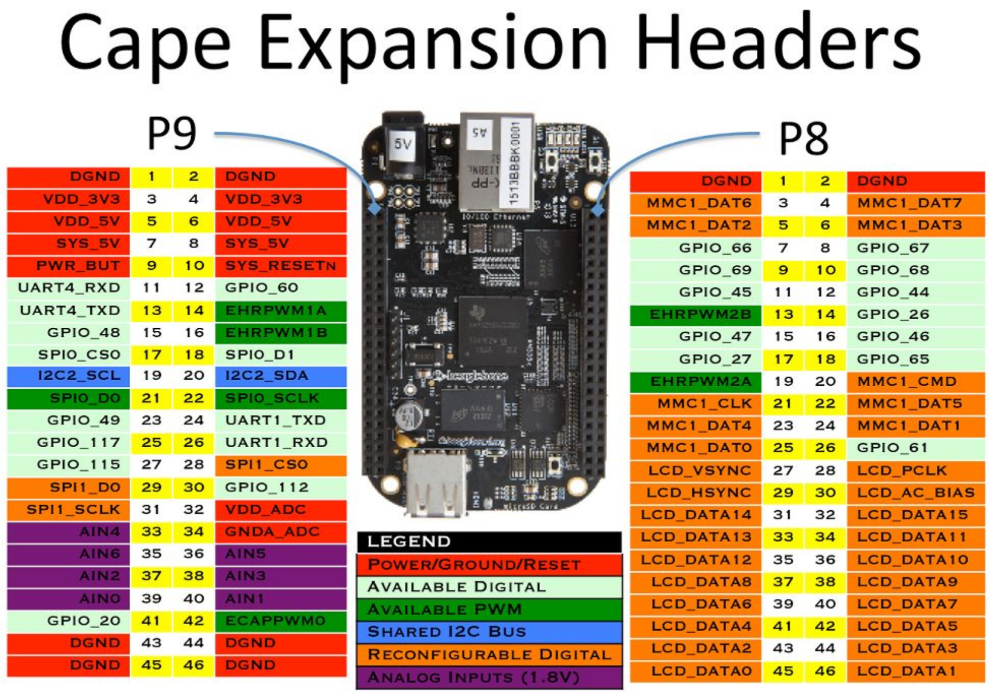
The kinematics of our handmade designer robot is composed by 5 rotational joints. Two actuators (q_1, q_5) are placed on the O_1 and O_5 axes , and are fixed to the support. As a parallel system, we can determine from (q_1, q_5) the corresponding position O_{xTerm} of the pen (direct geometric model), as well as we can calculate the articular position which drive to a desired O_{xTerm} (inverse geometric model).

Those functions are available in our open source code.

Material

BBB Controller

Beaglebone - Black



Encoders

E4P OEM Miniature Optical kit Encoder -US digital

Features

- Miniature size
- Push-on hub - spring loaded collet design
- Minimum shaft length of .375"
- Fits shaft diameters of .079" to .250"
- Accepts +/-.020" Axial shaft play
- Off-axis mounting tolerance of .010"
- 100 to 360 cycles per revolution (CPR)
- 400 to 1440 pulses per revolution (PPR)
- Single +5V supply

Motor Driver

MD1.3 2A Dual Motor Controller SKU DRI0002

Specifications

- The logic part of the input voltage: 6 ~ 12V
- Driven part of the input voltage Vs: 4.8 ~ 46V
- The logical part of the work current Iss: 36mA
- Drive part of the operating current Io: 2A
- Maximum power dissipation: 25W (T = 75 degree Celsius)
- Control signal input level:
- High level: 2.3V = Vin = Vss
- Low:-0.3V = Vin = 1.5V
- Operating temperature: -25 degree Celsius ~ +130 degree Celsius
- Drive Type: Dual high-power H-bridge driver
- Module Size: 47 mm × 53mm
- Module Weight: About 29g

5V regulator

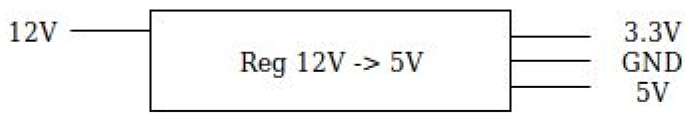
AMS - 1117

FEATURES

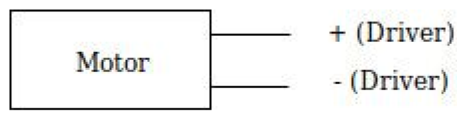
- **Three Terminal Adjustable or Fixed Voltages* 1.5V, 1.8V, 2.5V, 2.85V, 3.3V and 5.0V**
- **Output Current of 1A**
- **Operates Down to 1V Dropout**
- **Line Regulation: 0.2% Max.**
- **Load Regulation: 0.4% Max.**
- **SOT-223, TO-252 and SO-8 package available**

Connectics

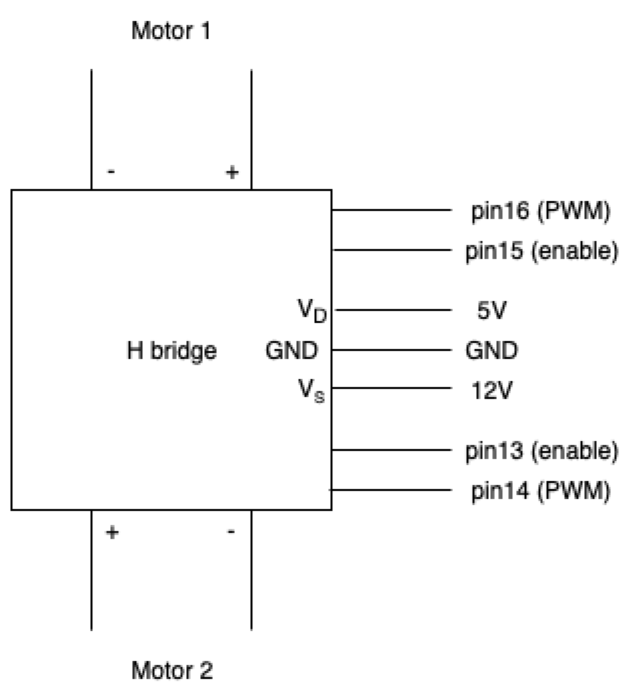
Voltage regulator :



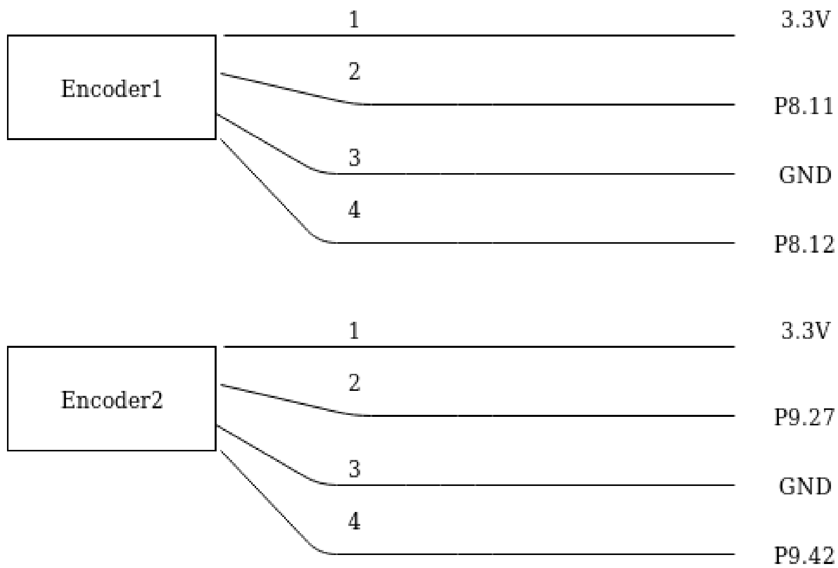
Motors :



Dual H-bridge (motor driver) :

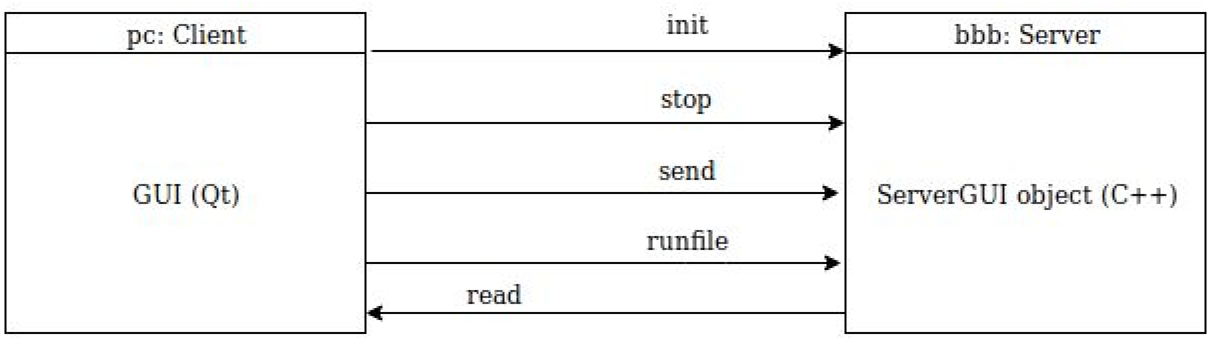


Encoders :


Encoder1
1
3.3V
2
P8.11
3
GND
4
P8.12
Encoder2
1
3.3V
2
P9.27
3
GND
4
P9.42

Graphical User Interface

Client / Server Communication



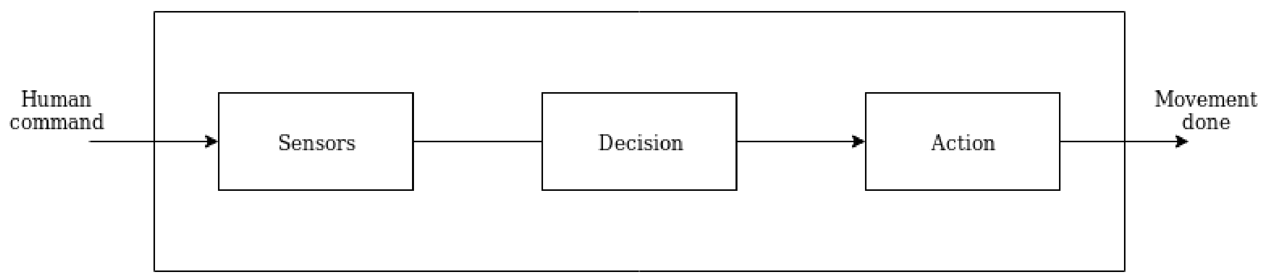
trame (char* / string)

Trame

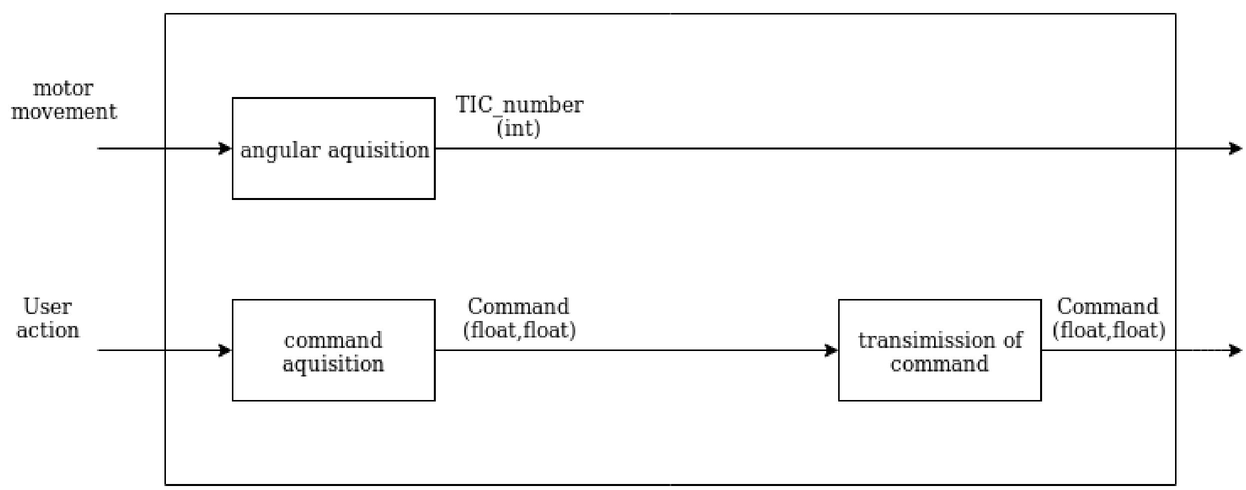
name	description	Trame (char* / string)
init	Initialize the system. The pen should be positioned on the frame origin marker.	“i:”
stop	Stop the bbb program running.	“e:”
send	Order the robot to move to the (xx.xx, yy.yy) position (cartesian space).	“s:xx.xx,yy.yy”
runfile	Run the path file given as argument.	“r:file_to_path.txt”
Connect	Get cartesian coordinates of the system.	“p:xx.xx,yy.yy”
Kill controller	Get joints coordinates of the system.	“a:q1,q5”
org	Set the encoder features.	“i:”

Software

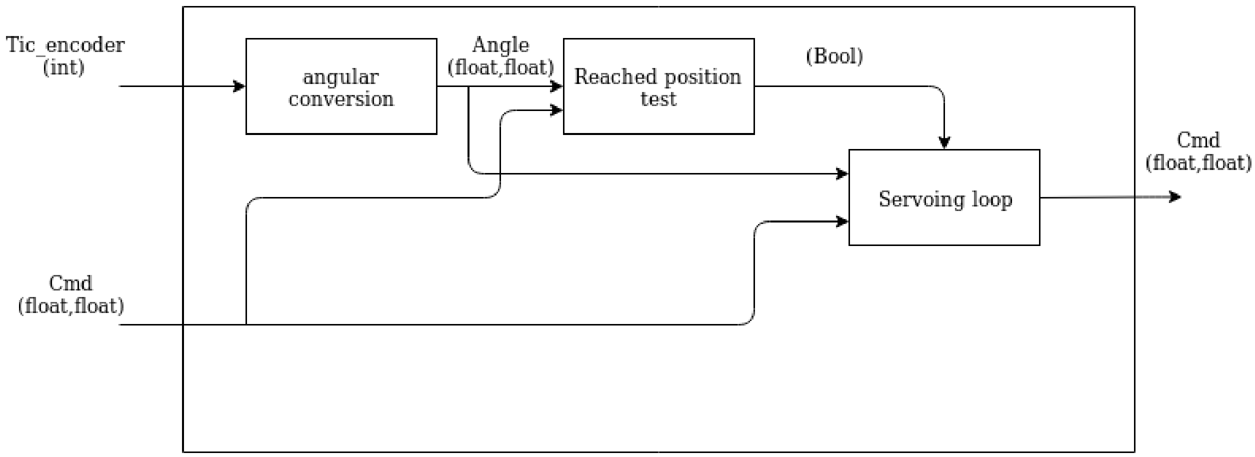
SADT



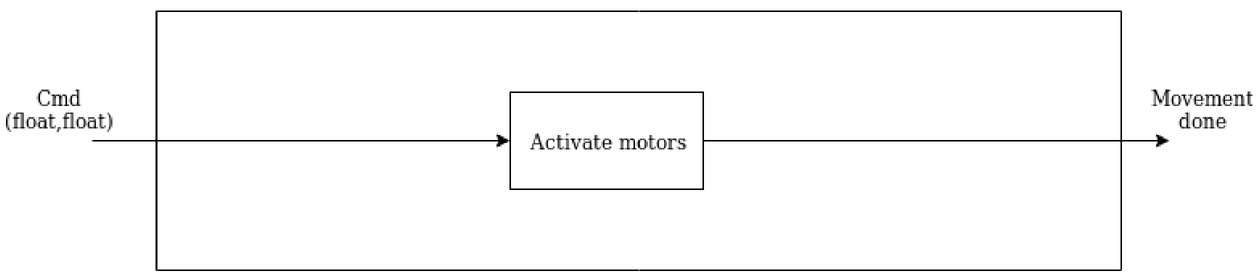
Sensors:



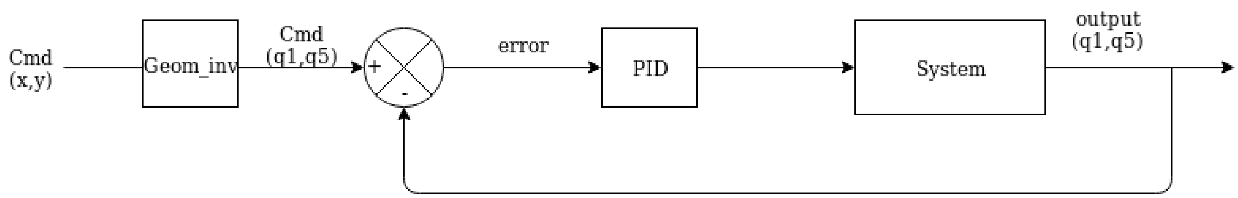
Decision:



Action:



Servoing



Working with Apollo 20

Getting started

1) Connect components as described in the Connectic part (p.7)
2) Launch the main python script with the following command on a new shell :
   ../Apollo20/interface$ python gui.py

The GUI will then appears on your workstation.

3) Put manually the robot effector on the origin marker, and clic on the **init** button.
4) Then clic on the org button.
5) Clic on the run controller button.
6) Finally, clic on the connect button.

The system is now initialized, you can start drawing with Apollo 20.

Interact through the Graphical User Interface

The GUI prints the current position of the system in the joint space (q_1, q_5), and in the cartesian space (x,y). It also allows the user to make the following tasks run :

- **Send** : order the robot to move to a specific position in the cartesian space
- **Runfile** : Run the path file which has been previously set on the bbb (controller/data.txt)