Automatic Illustrator

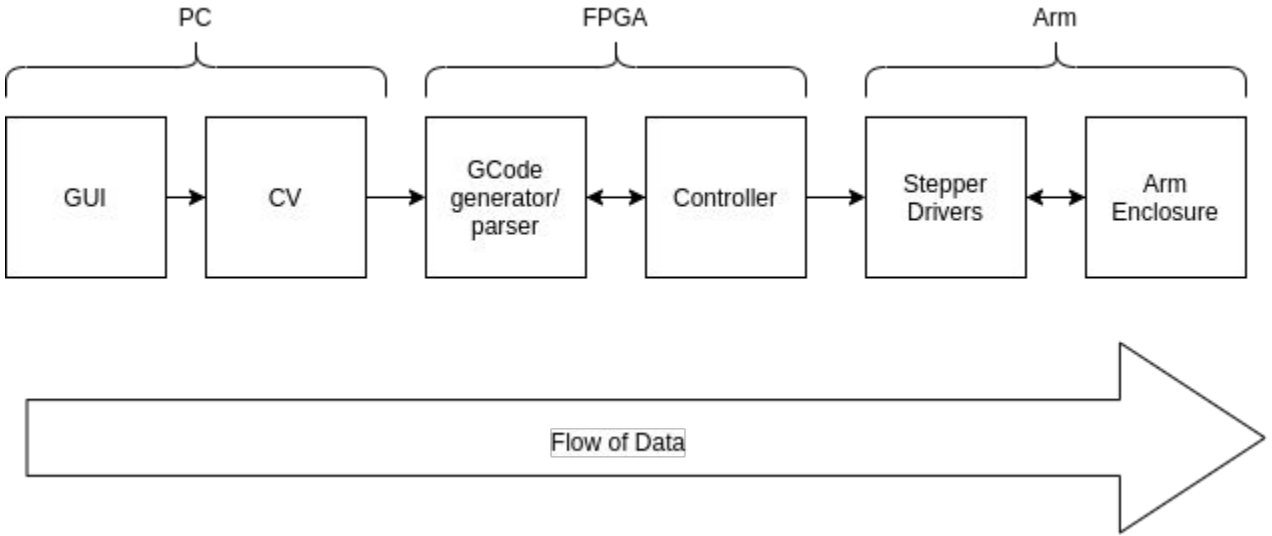
Tristan Thompson, Kyle Noble, Thomas Snyder

Project Overview

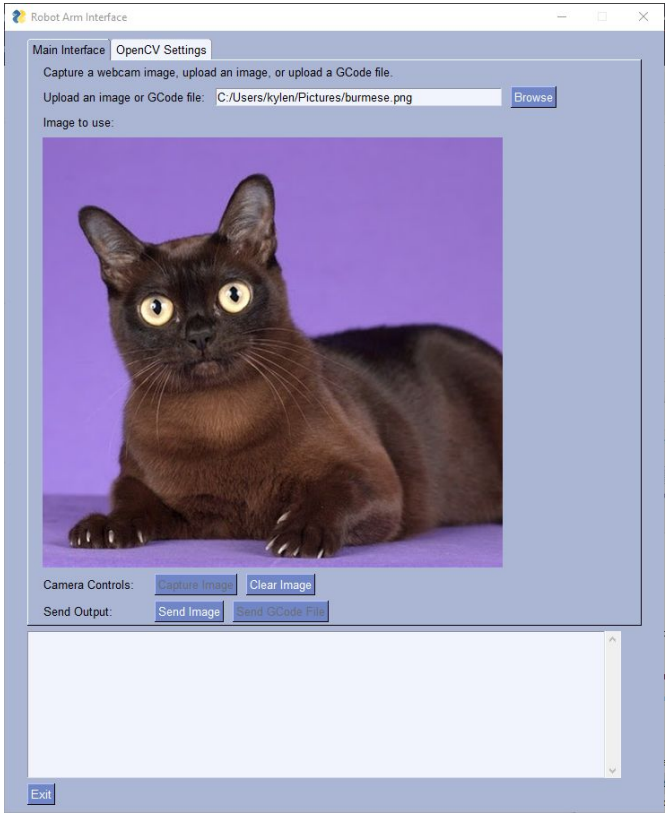
- A Robotic Arm that is able to draw on paper
- Uses OpenCV to generate GCode
- Parses GCode to control the movement of the arm
- Uses SCARA topology
- Uses hardware accelerated controller

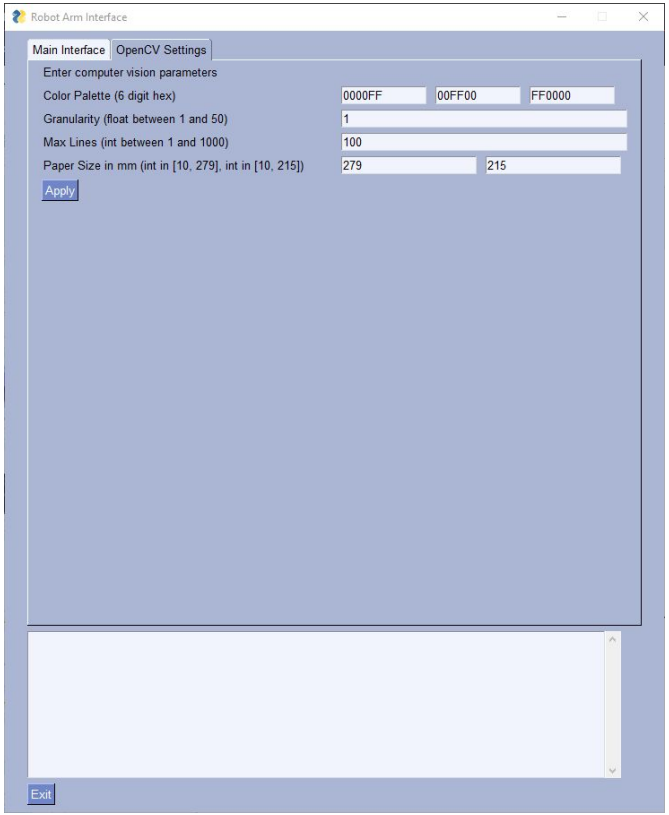
What is SCARA?

- Selective Compliance Articulated Robot Arm
- Two parallel joints that position an end effector at a point on the plane perpendicular to the rotational axes of the joints
- This plane is typically horizontal
- Position control is achieved by solving the inverse kinematics problem


PC
FPGA
Arm
GUI
CV
GCode generator/ parser
Controller
Stepper Drivers
Arm Enclosure
Flow of Data

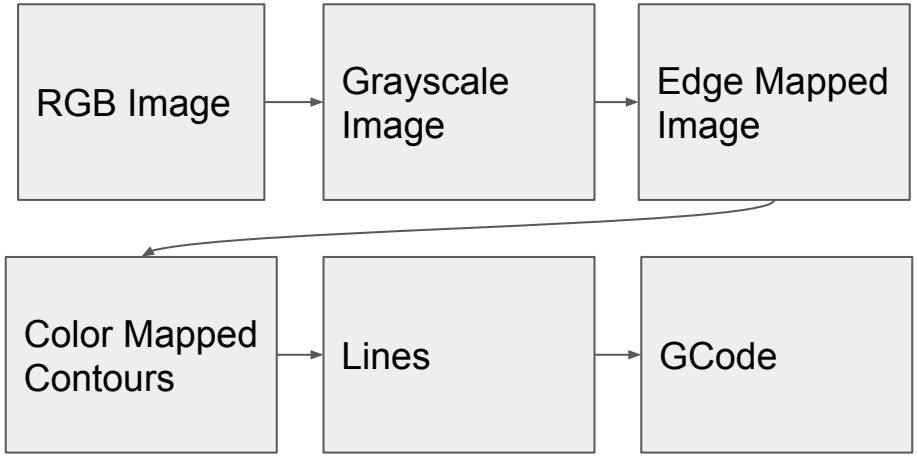
GUI


Robot Arm Interface
Main Interface
OpenCV Settings
Capture a webcam image, upload an image, or upload a GCode file.
Upload an image or GCode file:
C:/Users/kylen/Pictures/burmese.png
Browse
Image to use:
Camera Controls:
Capture Image
Clear Image
Send Output:
Send Image
Send GCode File
Exit


Robot Arm Interface
Main Interface
OpenCV Settings
Enter computer vision parameters
Color Palette (6 digit hex)
0000FF
00FF00
FF0000
Granularity (float between 1 and 50)
1
Max Lines (int between 1 and 1000)
100
Paper Size in mm (int in [10, 279], int in [10, 215])
279
215
Apply
Exit

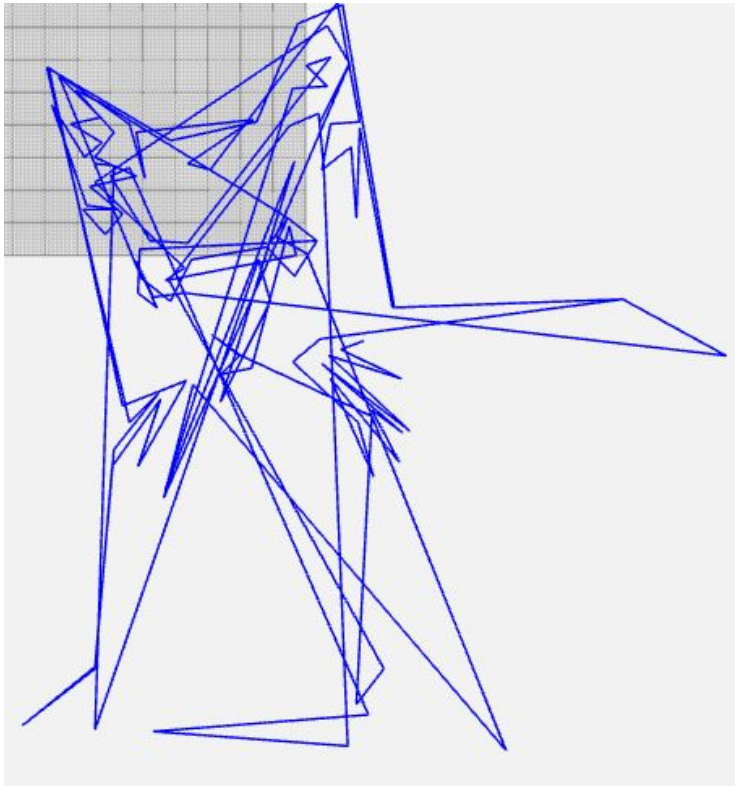
CV

Transformation Summary

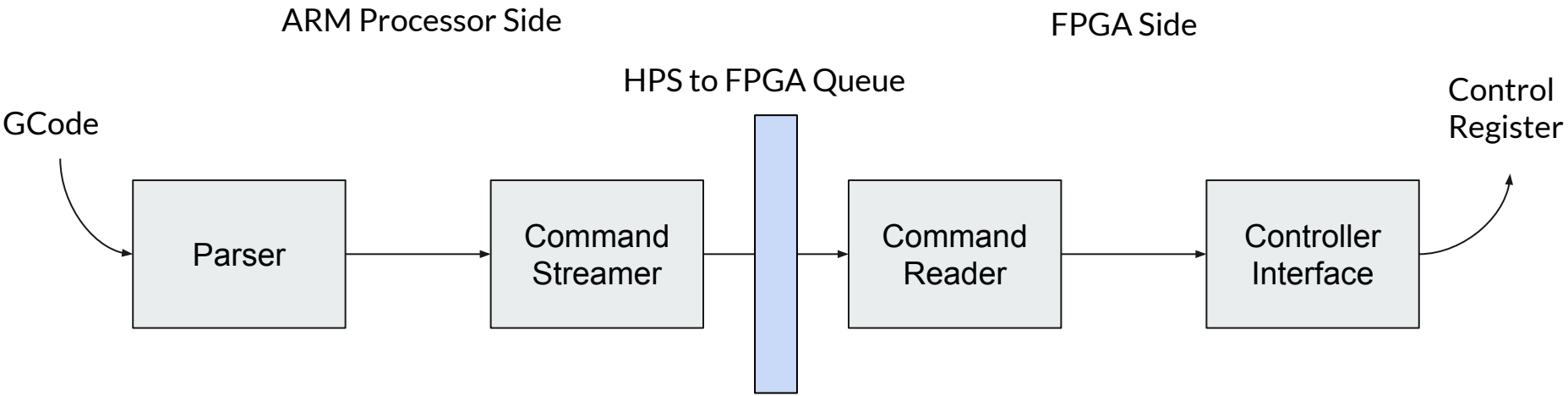


Resultant GCode

```
G90
G21
M72
G01 X152.01499527801997 Y309.3434866220285
M72
M72
G01 X116.38845965148433 Y303.3664751277756
M72
M72
G01 X116.14275940578408 Y308.8837165070859
M72
G01 X118.84546210848679 Y308.6538314496147
M72
G01 X115.40565866868334 Y249.1136015645572
M72
G01 X116.14275940578408 Y246.5848659323733
M72
G01 X139.48428274730742 Y298.3090038634078
M72
G01 X141.20418446720913 Y293.7113027139825
M72
```



GCode Generator/Parser



Command Format

- Command is a value from 0-8, representing G00, G01, G20, G21, G90, G91, M2, M6, M72
- X Argument, Y Argument is a 14 bit number in units of 2^{14} bits per maximum paper size in inches or mm

GCode Interpolation

Controller

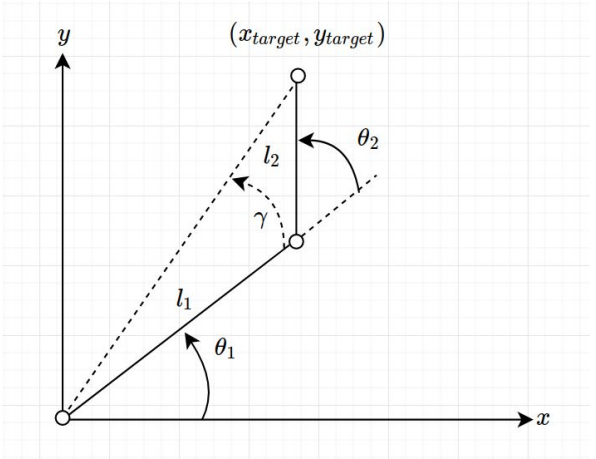
Interfacing the GCode with the stepper motors.

Controller: Inverse Kinematics

Translating the points from the G-Code parser to joint angles.

Inverse Kinematics: Calculating θ_2

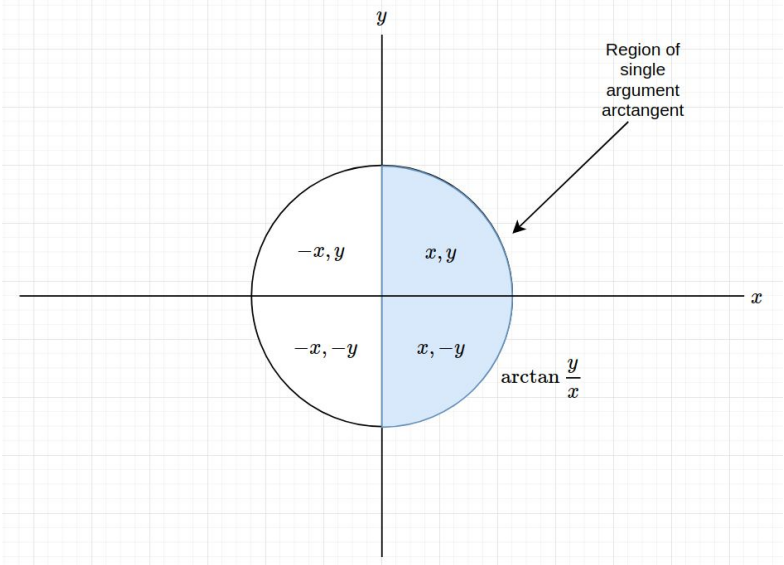
$$\cos\theta_2 = \frac{x_{target}^2 + y_{target}^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

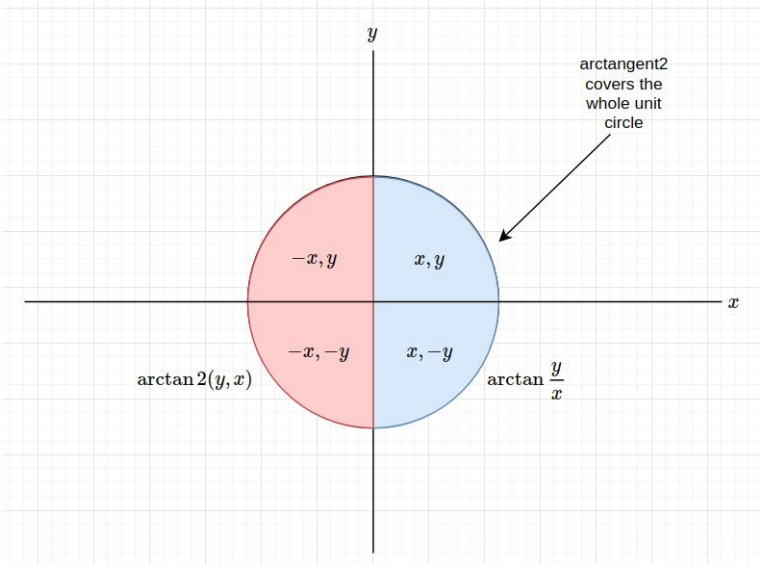


$$\sin\theta_2 = \sqrt{1 - \cos^2\theta_2}$$

$$\theta_2 = \arctan 2(\sin\theta_2, \cos\theta_2)$$

What is the two argument arctangent?

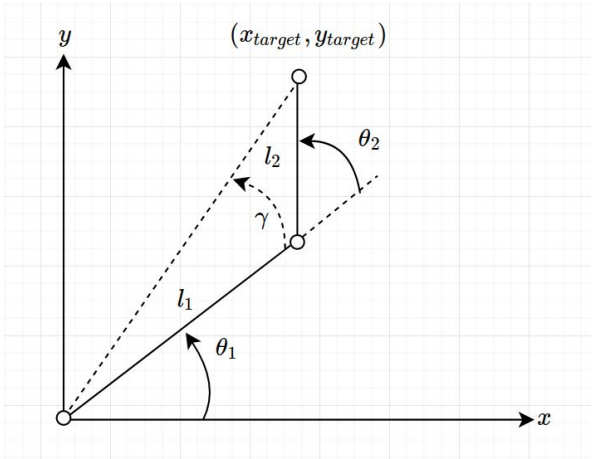




Finding θ_1

$$k_1 = l_1 + l_2 * \cos\theta_2$$

$$k_2 = l_2 * \sin\theta_2$$



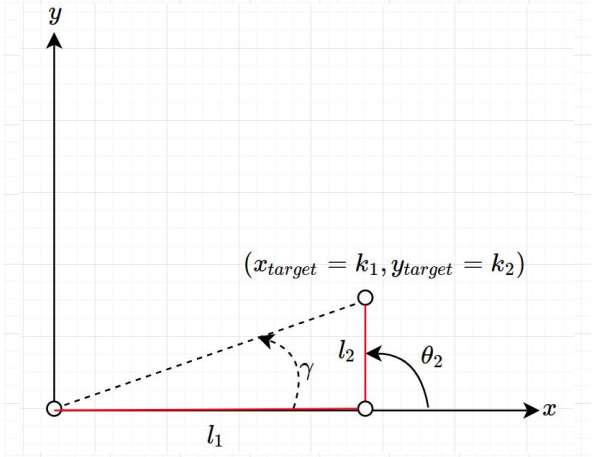
$$\gamma = \arctan 2(k_2, k_1)$$

$$\theta_1 = \arctan 2(y, x) - \gamma$$

Finding θ_1

$$k_1 = l_1 + l_2 * \cos\theta_2$$

$$k_2 = l_2 * \sin\theta_2$$



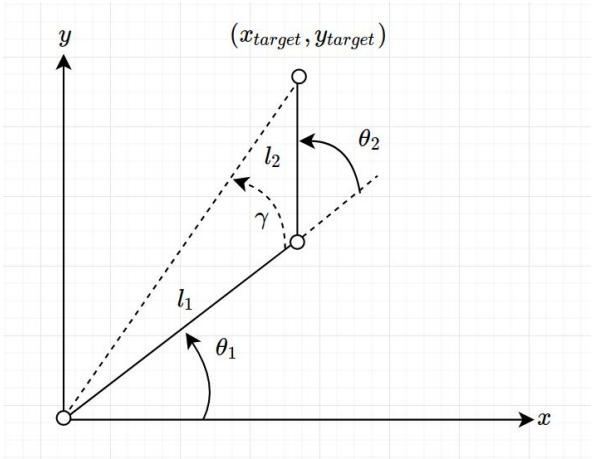
$$\gamma = \arctan 2(k_2, k_1)$$

$$\theta_1 = \arctan 2(y, x) - \gamma$$

Finding θ_1

$$k_1 = l_1 + l_2 * \cos\theta_2$$

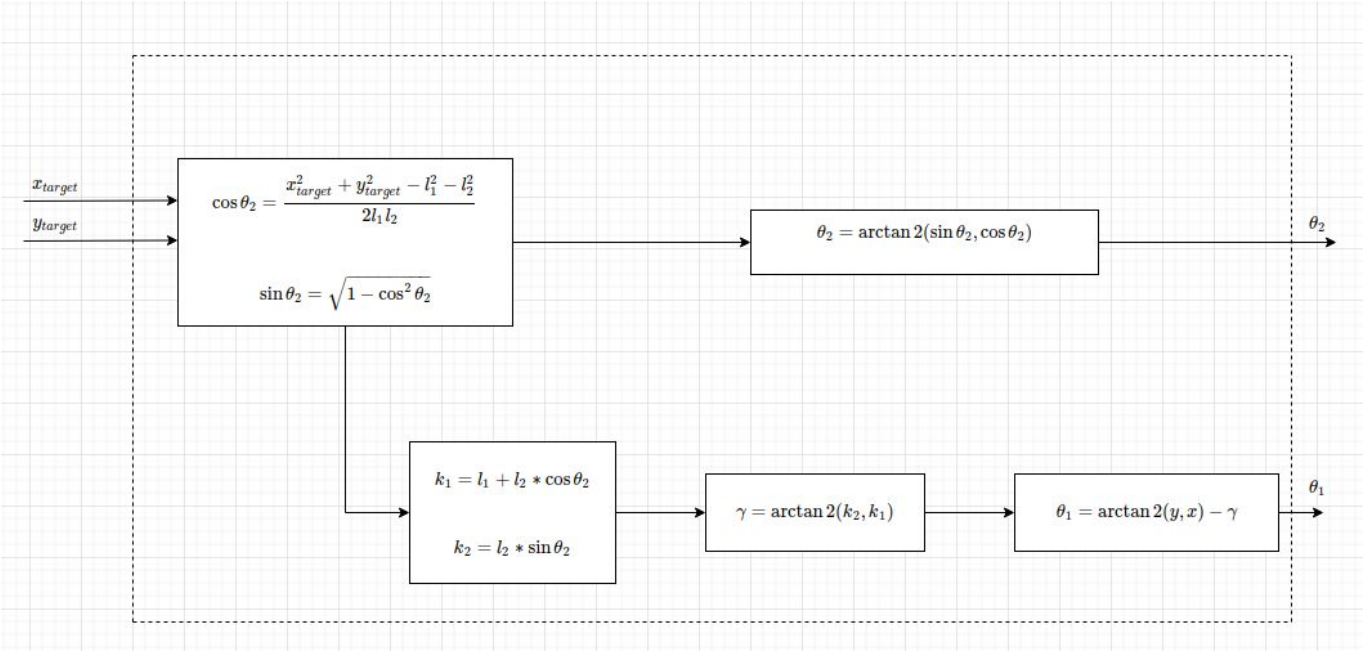
$$k_2 = l_2 * \sin\theta_2$$



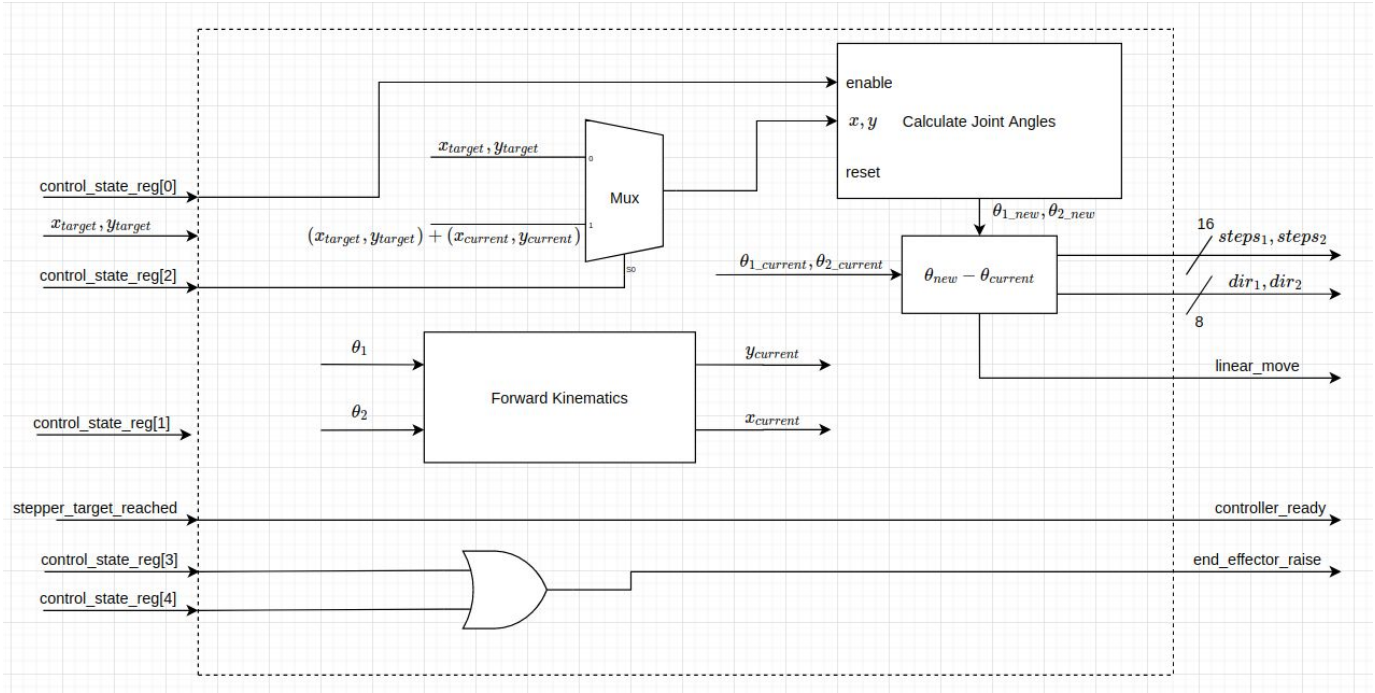
$$\gamma = \arctan 2(k_2, k_1)$$

$$\theta_1 = \arctan 2(y, x) - \gamma$$

Inverse Kinematics: Mathematical Overview



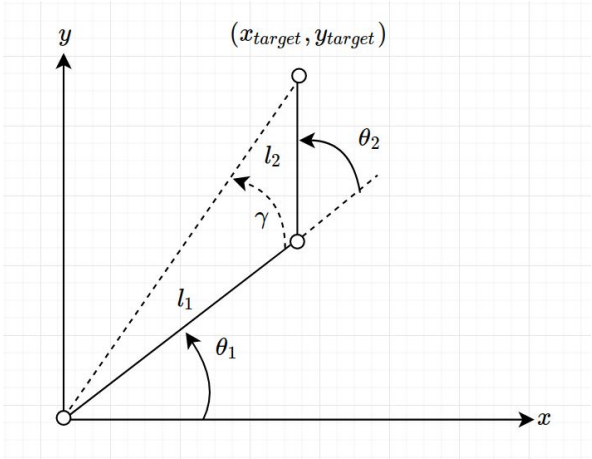
Controller: Overview



Forward Kinematics

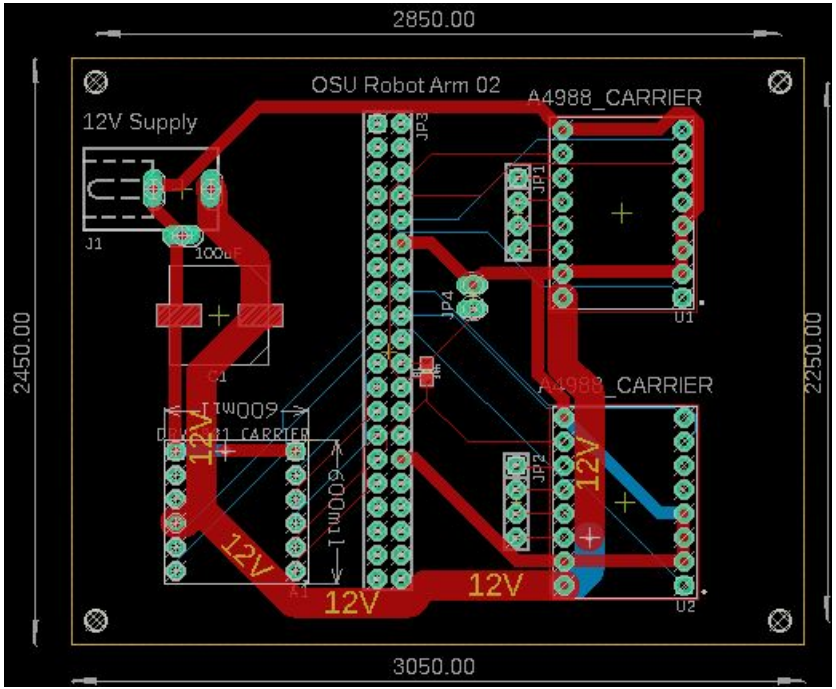
$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$



Stepper Motor Drivers

- Uses 12V power supply.
- Each stepper motor uses an A4988 stepper motor driver carrier from Pololu.
- DC motor uses the DRV8801 DC motor driver from Pololu.
- These components were selected to match the 3.3V logic of the DE1-SoC, to handle a 12V power supply, and for their current limiting capabilities.
- I/O delivered through 40 pin GPIO on DE1-SoC and header pins.



Enclosure and Arm

Housing the components and the drawing the images.

Challenges

- FPGA firmware development
- Hardware-software compatibility
- Hardware resolution
- Mechanical issues
  - PLA strength
  - Belt tensioning mechanism
  - Noncompliant end effector

Next Steps

- Planetary gearbox and micro-stepping for higher resolution
- More belt tension
- Sturdier arm material/design
- Hold the utensil with a spring system that can vary in height
- Closed loop control
- Convert all FPGA calculations to 64-bit floating point to reduce rounding errors