Two-Axis Robotic Arm

Arthur Kichatov, Jialiang Wang, Mingqian Xu, Tianhao Lin

Project Description

For this project, the group will be asked to develop a robotic arm that can draw pictures on a 8.5” x 11” sheet of paper. This project involves embedded design,control systems, and MATLAB or Python scripting.

System Requirements

- **Customer Requirement:** The system should be fast.
  Engineering Requirement: The system must be able to draw faster than 4 inches per second.

- **Customer Requirement:** The system must be accurate.
  Engineering Requirement: The system must be able to draw a 10 inch straight line +/- .25 inch. This includes both the overall length of the line and ensuring the line does not vary more than .25 inches of being perfectly straight.

- **Customer Requirement:** The system needs to be inexpensive and manageable to manufacture.
  Engineering Requirement: The robotic arm will use a SCARA topology, with two rotating joints to control arm actuation.

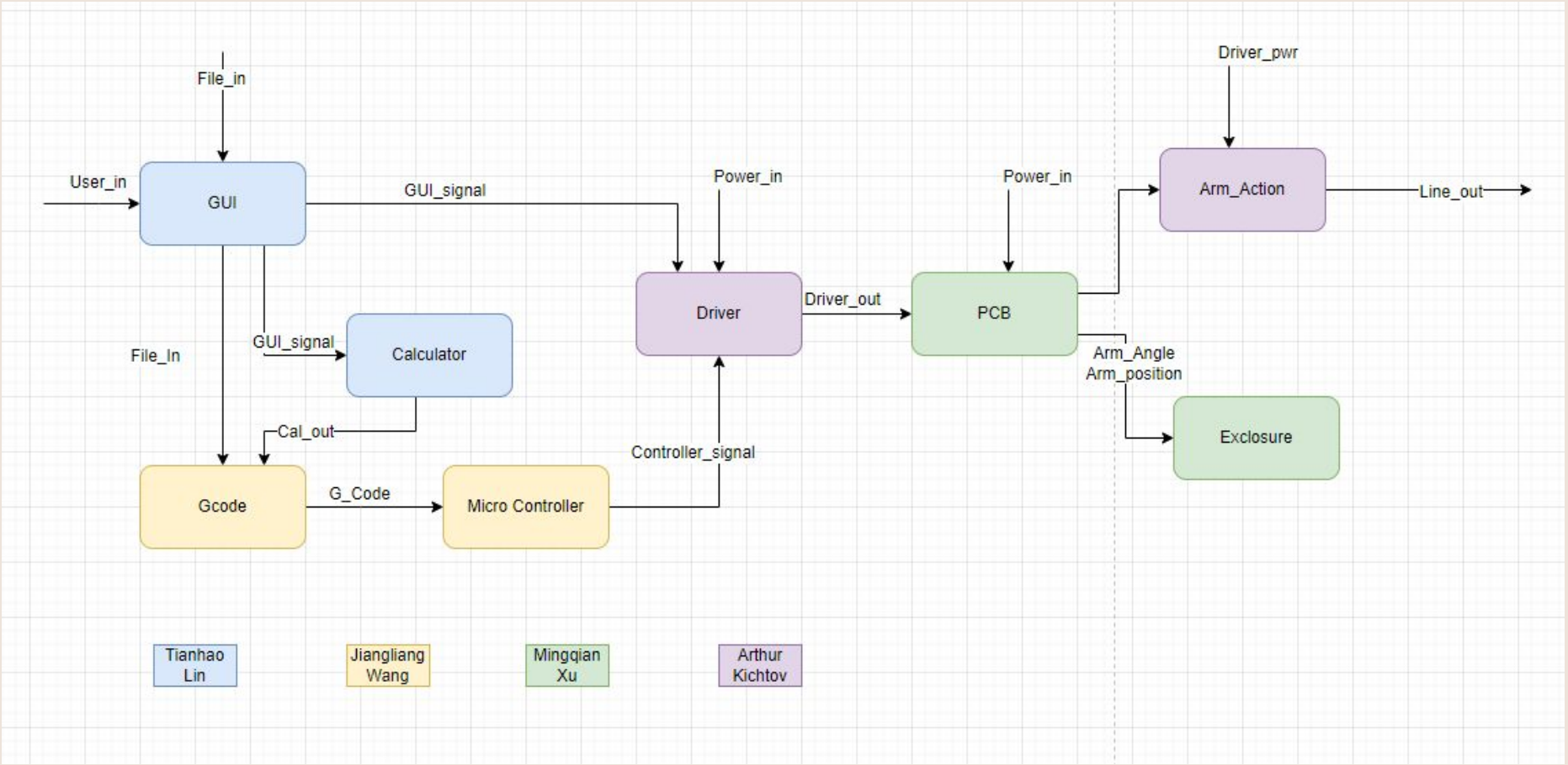
- **Customer Requirement:** The system must have a commonly known interface.
  Engineering Requirement: Controlling commands will be input as G-code commands. These commands must be made available within the Python or MATLAB GUI.G0, G1, G90, G91, G20, G21, M2, M6, M72.

- **Customer Requirement:** The system must use different types of writing tools.
  Engineering Requirement: Upon receiving an M6 command the machine operator must be able to mount a crayon, pen, or pencil within 15 seconds.

- **New Customer Requirement:** The system will allow for manual control.
  Engineering Requirement: By choosing the manual option, the arm can be controlled manually through buttons in the user interface.

- **New Customer Requirement:** The system can write math operations automatically.
  Engineering Requirement: The user will input a math equation into the user interface and a vector map will be generated for the arm to use

Blocks


File_in
User_in
GUI
GUI_signal
Power_in
Power_in
Driver_pwr
Arm_Action
Line_out
Driver
Driver_out
PCB
File_In
GUI_signal
Calculator
Arm_Angle
Arm_position
Exclosure
Cal_out
Controller_signal
Gcode
G_Code
Micro Controller
Tianhao Lin
Jiangliang Wang
Mingqian Xu
Arthur Kichtov

Interface

Interface and properties		
Name	Type	Parameter
File_in	Picture files (*.jpg or *.png)	Content in the file
User_in	Code and text	Matlab (*.m)
GUI_signal	Analog signal	V range: 0-12V I range: 100mA-1200mA
Cal_out	Analog signal (a float number)	Floating point
G_code	Code (*.m)	Geometric code
Controller_signal	Analog signal	V range: 0-12V I range: 100mA-1200mA
Power_in	DC power (12 V)	V range: 0-12V I range: 100mA-1200mA
Driver_out	Analog signal	V range: 0-12V I range: 100mA-1200mA
Driver_pwr	DC power (12 V)	V range: 0-12V I range: 100mA-1200mA
Arm_angle and arm_position	Motor action	V range: 0-12V I range: 100mA-1200mA
Line_out	Arm move	8.5” * 11” paper draw

Timeline

Week 12: Determine Project

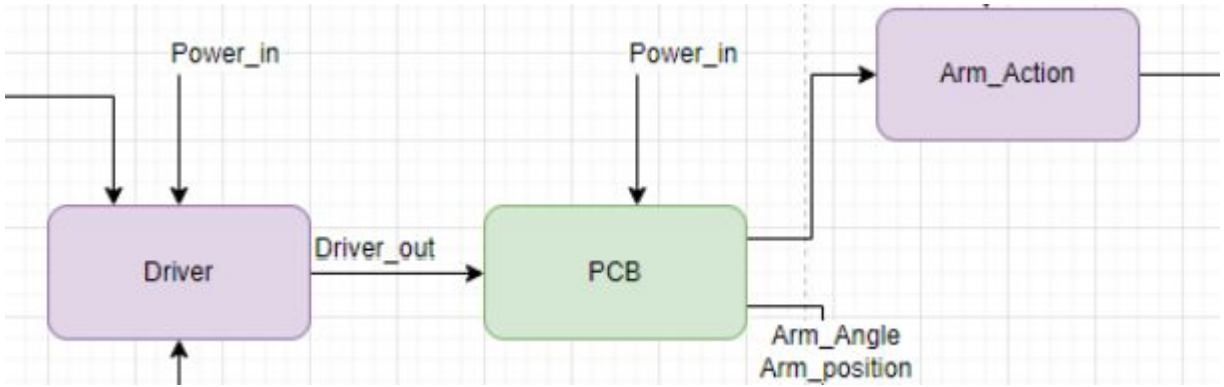
Week 14: First Block Check-Off

Week 17: Second Block Check-Off

Week 19: Final Check-Off

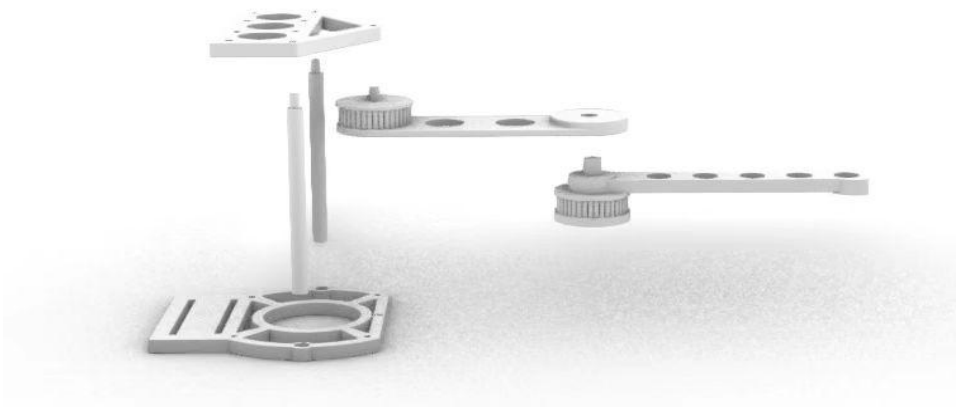
Motor Block

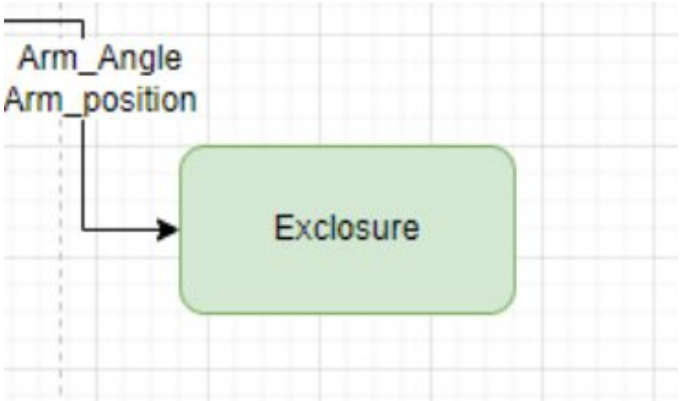
- Uses Microcontroller and stepper motor Drivers
- Built on a single PCB
- Takes G-Code inputs and runes the motors using the drivers
- Reads GUI input and moves the arm



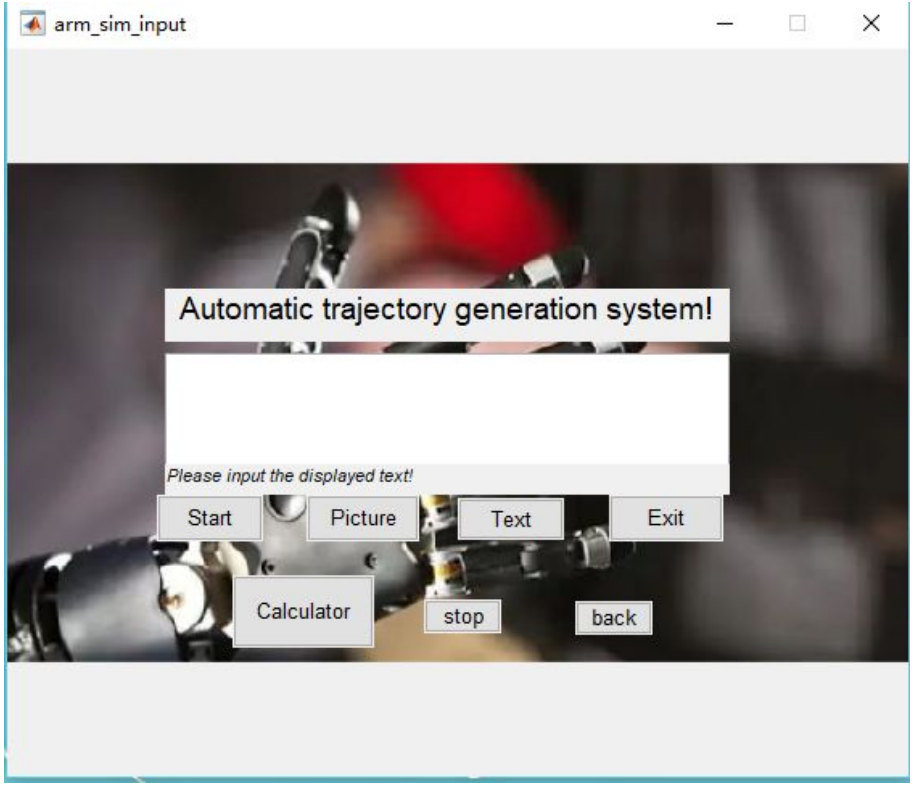
Enclosure

- 3D printed plastic material for arms
- Honeycomb design arm structure for lightness

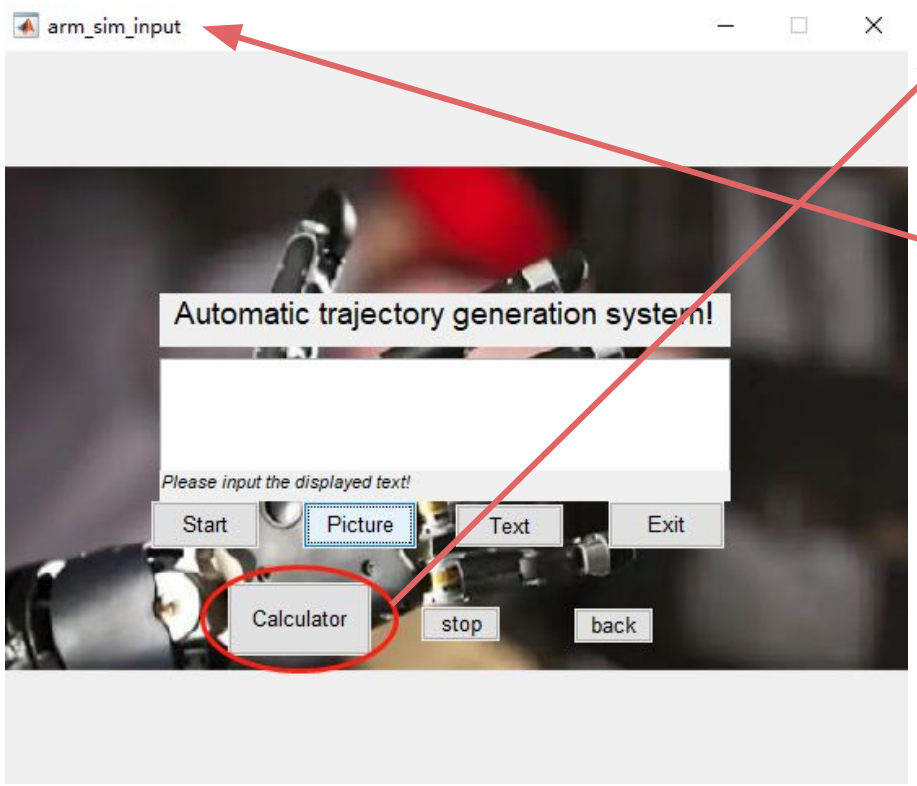


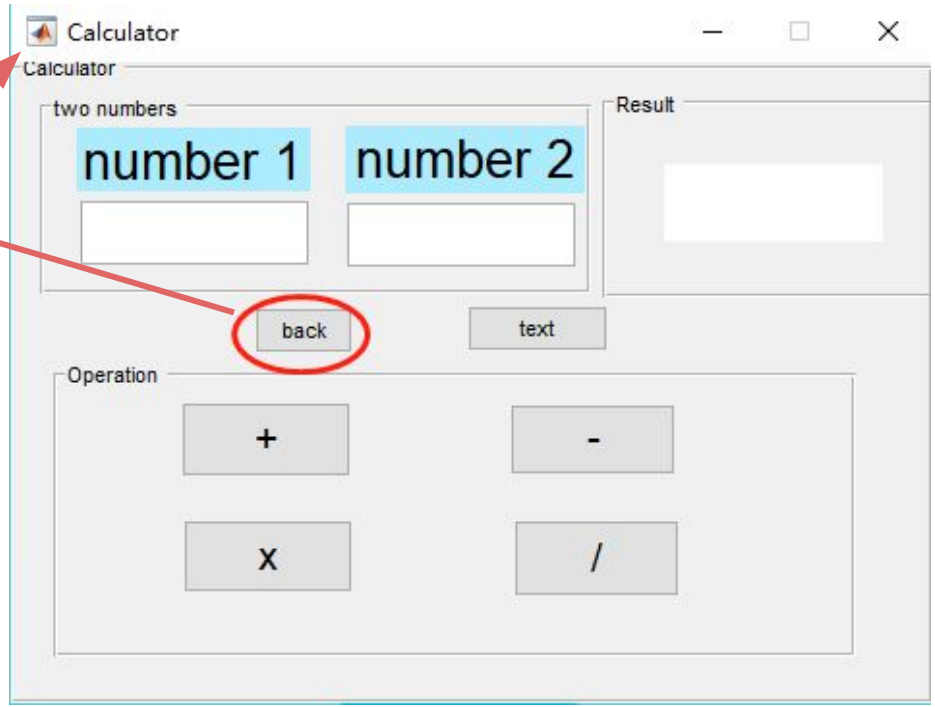


Graphical User Interface

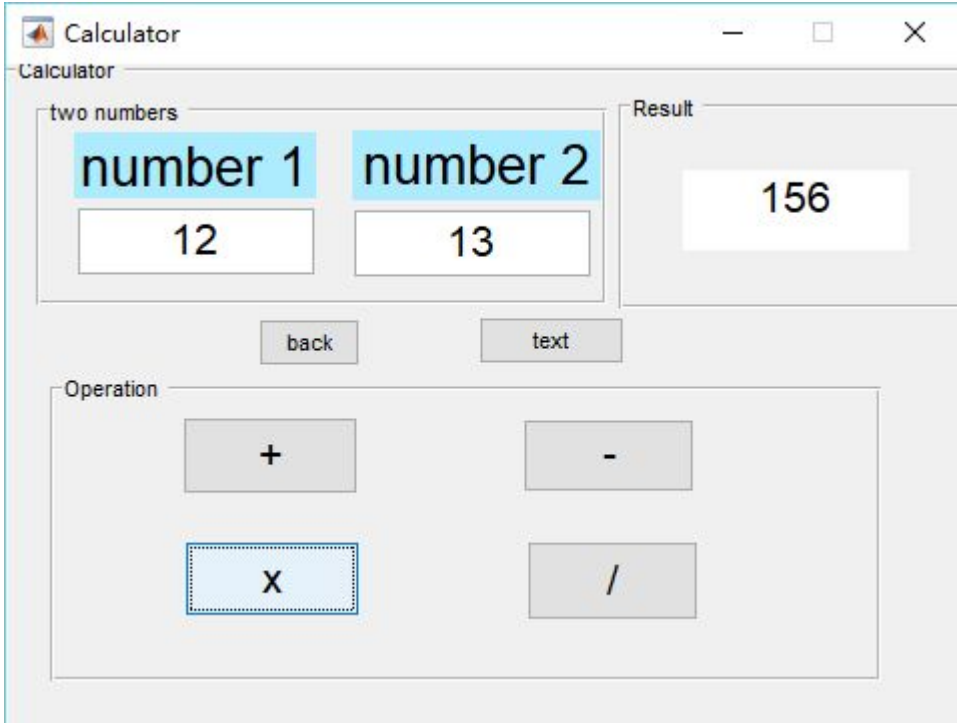


GUI & Calculator

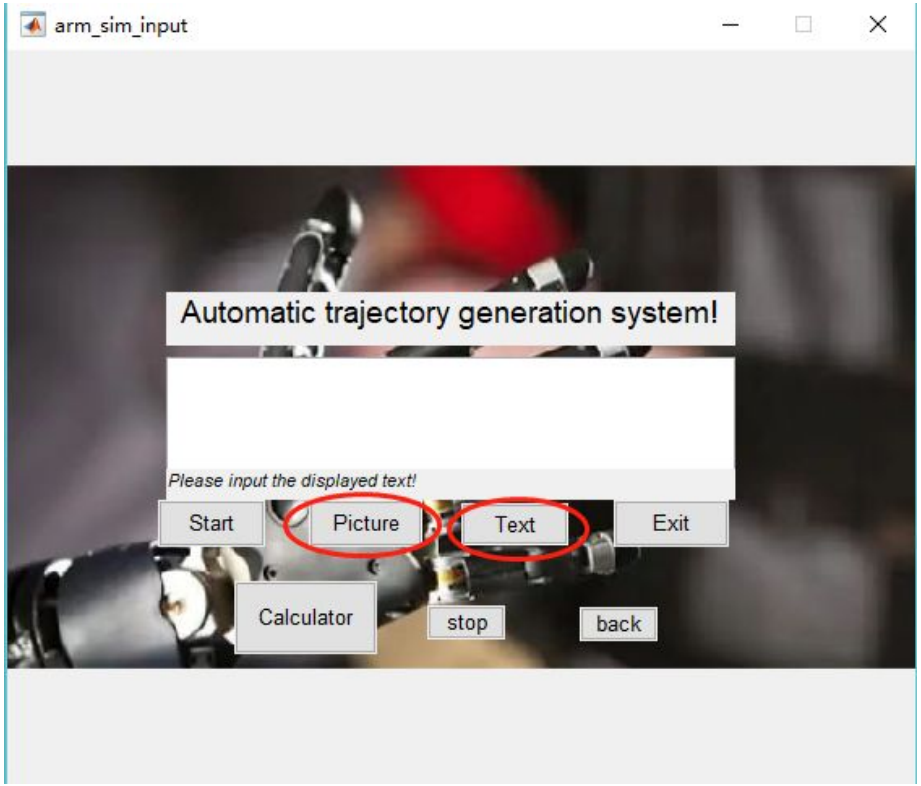




Calculator


Calculator
Calculator
two numbers
number 1
12
number 2
13
Result
156
back
text
Operation
+
-
x
/

GUI -> Gcode


arm_sim_input
Automatic trajectory generation system!
Please input the displayed text!
Start
Picture
Text
Exit
Calculator
stop
back