Interface Definitions

Robot Arm (02)

Thomas Snyder, Kyle Noble, Tristan Thompson

Functional Block Diagram

Block Name	Description	Input Specifications	Output Specifications
GUI	User interface that allows the user to give an image or G-Code file to be sent to the arm. Also shows the state of the Command Parser.	**Name:** user_GUI **Description:** Input from the user that selects which input method to use and then graphical interfaces to control input parameters/upload files. The user can select a G-Code file to be sent directly to the G-Code parser or capture an image on the webcam and send it to the CV block along with the desired parameters for generating the contours. **Name:** Parser_GUI **Description:** The state of the command parser to show on the GUI. Includes information about what color to switch to next on tool changes and the arm	**Name:** GUI_CV **Description:** Image from webcam source. **Name:** GUI_Parser **Description:** G-Code file. Must be a .gcode file with a file size no greater than 512MB. **Name:** GUI_CV **Description:** Image from webcam source. Should be between 320 x 480 and 3840 x 2160 resolution. **Name:** CV_Color **Description:** What colors need to be swapped in when switching tools. Colors are a RGB tuple ranging from 0 to 255. Three colors will be sent as a list of tuples.

		state.	**Name:** Granularity **Description:** How dense the gcode lines are. Must be a float between 1 and 50 **Name:** Max_Countours **Description:** Maximum number of contours sampled to create lines. Must be an integer between 1 and 1000 **Name:** Paper_Size **Description:** Size of paper in millimeters. Must be a tuple of two integers. Width between 10 and 279. Height between 10 and 215
Computer Vision	Processes image data into g-code text using color edge detection.	**Name:** GUI_CV **Description:** Image from webcam source. Should be between 320 x 480 and 3840 x 2160 resolution. **Name:** CV_Color **Description:** What colors need to be swapped in when switching tools. Colors are a RGB tuple ranging from 0 to 255. **Name:** Granularity **Description:** How dense the gcode lines are. Must be a float between 1 and 50 **Name:** Max_Countours **Description:** Maximum number of	**Name:** CV_Parser **Description:** G-Code text in a file that is less than 512 MB. The text describes the lines that the CV has extracted from the image.

		contours sampled to create lines. Must be an integer between 1 and 1000 **Name:** Paper_Size **Description:** Size of paper in millimeters. Must be a tuple of two integers. Width between 10 and 279. Height between 10 and 215	
Command Parser	Syntax checks and parses G-Code into commands for the FPGA. Communicates with FPGA through shared memory.	**Name:** CV_Parser **Description:** Valid G-Code text containing the G-Code commands specified in the customer requirements. Arguments to commands must not exceed the size of the paper. **Name:** ctrlr_parser **Description:** The controller will output a status signal that consists of a single bit that is pulsed upon completion of parser commands. **Name:** GUI_parser **Description:** The GUI passes a callback function to the parser that will later be used to display the arm state and the color that should be placed in the grasper.	**Name:** Parser_ctrlr **Description:** The command is a 5-bit bus representing the G-Code instruction. Arg1 and Arg2 will be 14-bit integer values that are interpreted as signed when in relative mode and unsigned when in absolute mode. See parser_ctrlr input specification in the controller block for integer specifications. For the command: **Bit 0** represents a movement **Bit 1** represents units (1 = inches, 0 = millimeters) **Bit 2** represents positioning method (0 = absolute, 1 = relative) **Bit 3** represents raising the tool **Bit 4** represents a tool change

			Name: Parser_GUI **Description:** The state of the command parser to show on the GUI. Includes information about what color to switch to next on tool changes and the arm state.
Controller	A multifunctional system that interprets G-codes and outputs the necessary physical movements or actions. For specific G-codes, move the arm or change the utensil.	**Name:** parser_ctrlr **Description:** Based on the G-code received, the parser will set the appropriate bits in the controller state register and pass two 14-bit arguments (X, Y position). The controller state register will be a 5-bit register and the arguments will be integer values in the unit conversion set by the "units" G-code and reflected in the state register. If the units are in inches, each integer represents $22 / 2^{14}$ inches. **Name:** motor_ctrlr **Description:** Receive a single bit indicating whether the stepper motor driver has reached the given target (1 means target reached). Also receives a single bit indicating the state of the end effector (0 means up and 1 means down).	**Name:** ctrlr_motor **Description:** The controller will output two 8-bit integers and two one-bit values. The integers represent the number of steps the steppers must make to get to the specified position, which must be within ***one half of a step*** to the necessary step number. The direction bits will specify 1 as positive (counterclockwise) rotation and 0 as negative (clockwise) rotation. For the first joint ("shoulder"), the angle will be in the frame of the world. For the second joint ("elbow"), the angle will be in the world frame transposed to have the origin at its joint. Lastly, the controller will output a single bit that represents the desired state of the end effector (0 means up, 1 means down). There will also be a data_ready bit output by the controller to the motors that indicates the output data is ready to be read. **Name:** ctrlr_parser

			Description: The controller will output a status signal that consists of a single bit that is pulsed upon completion of parser commands.
Stepper Motor Driver	The machine interface with the actuators. This controls all of the actuators on the robot, translating electrical signals into motion and moving the actuators to desired positions based on commands from the controller. This block includes the motors.	**Name:** ctrlr_motor **Description:** The controller provides two integers representing the number of steps that the driver needs to move the motors. It will also provide two, one-bit values, each representing a direction for the motor to travel (for the definition of these values see ctrlr_motor output specification in controller block). This will require calibration in the world frame (starting the joints in specific orientations). New target data is only read on the rising edge of the “new_in” data bit.	**Name:** motor_encl **Description:** Actuates the motors via a PWM signal to move the enclosure to the desired positions. **Name:** motor_ctrlr **Description:** A single bit that indicates when the servo driver has reached the target coordinate. Sends another bit that indicates the position that the end effector is in.
Enclosure	The physical construction of the Robot and its supporting circuitry.	**Name:** motor_encl **Description:** The motors will rotate the joints of the arm via a belt and pulley system. The motor must have mounting points for two NEMA 17 stepper motors **Name:** fpga_encl **Description:** The enclosure must have 4 mounting points for the DE1-SOC control board.	**Name:** encl_paper **Description:** The enclosure holds a utensil that can write on the paper and withstand 3 Newtons of lateral force for 5 seconds at the tip of the utensil. When the belts are fully tensioned, the arm will deflect no more than 1 cm over the whole length. When moving and halting, the arm will have backlash of no more than +/- 0.5 cm per section, with total backlash of no more than +/- 1 cm at the end effector.

		The enclosure must have cutouts exposing USB and VGA I/O **Name:** pcb_encl **Description:** The enclosure must have 4 mounting points for the stepper driver PCB and options for cable management	

Black Box Block Diagram

Interface Name	Description
power_arm	The arm will be powered via 120v AC that will be stepped down to 12v for the FPGA and load supply on the stepper motor drivers and the FPGA will provide 3.3v VDD logic voltage to the stepper motor drivers.
user_arm	The user will take a picture using a webcam to be converted to G-code or the user will provide the G-code to draw using a monitor and mouse.
arm_paper	The arm will draw an image specified by the user onto the paper that is provided with the writing utensil provided by the user.
arm_user	The arm will notify the user when utensils must be changed (for colors), and will also provide feedback for improper use of the GUI.

Hardware Associations

Hardware	**Associated Blocks**
FPGA Coprocessor	1. GUI 2. Computer Vision 3. Command Parser 4. Controller 5. Stepper Motor Driver a. PID loops and feedback
Physical Arm	1. Enclosure 2. Stepper Motor Driver a. Interface between FPGA and motors

Note: The FPGA receives power, and all blocks on the FPGA will be powered by said FPGA. This is difficult to clearly show on the block diagram. The only hardware that receives power from the external power_arm interface are the stepper motor driver hardware and the FPGA.