IEEEDuino Student Contest 2024
Design Stage

Team Name:

IEEE Section: EGYPT

Team Members:

- Abdallah Ashraf
- Nour Mamdouh
- Mohamed Ashraf

University or Student Branch Name: CAIRO UNIVERSITY STUDENT BRANCH

Project Title: FUZZCAR

Project Scope, Purpose and Background

The purpose of this project is to address...

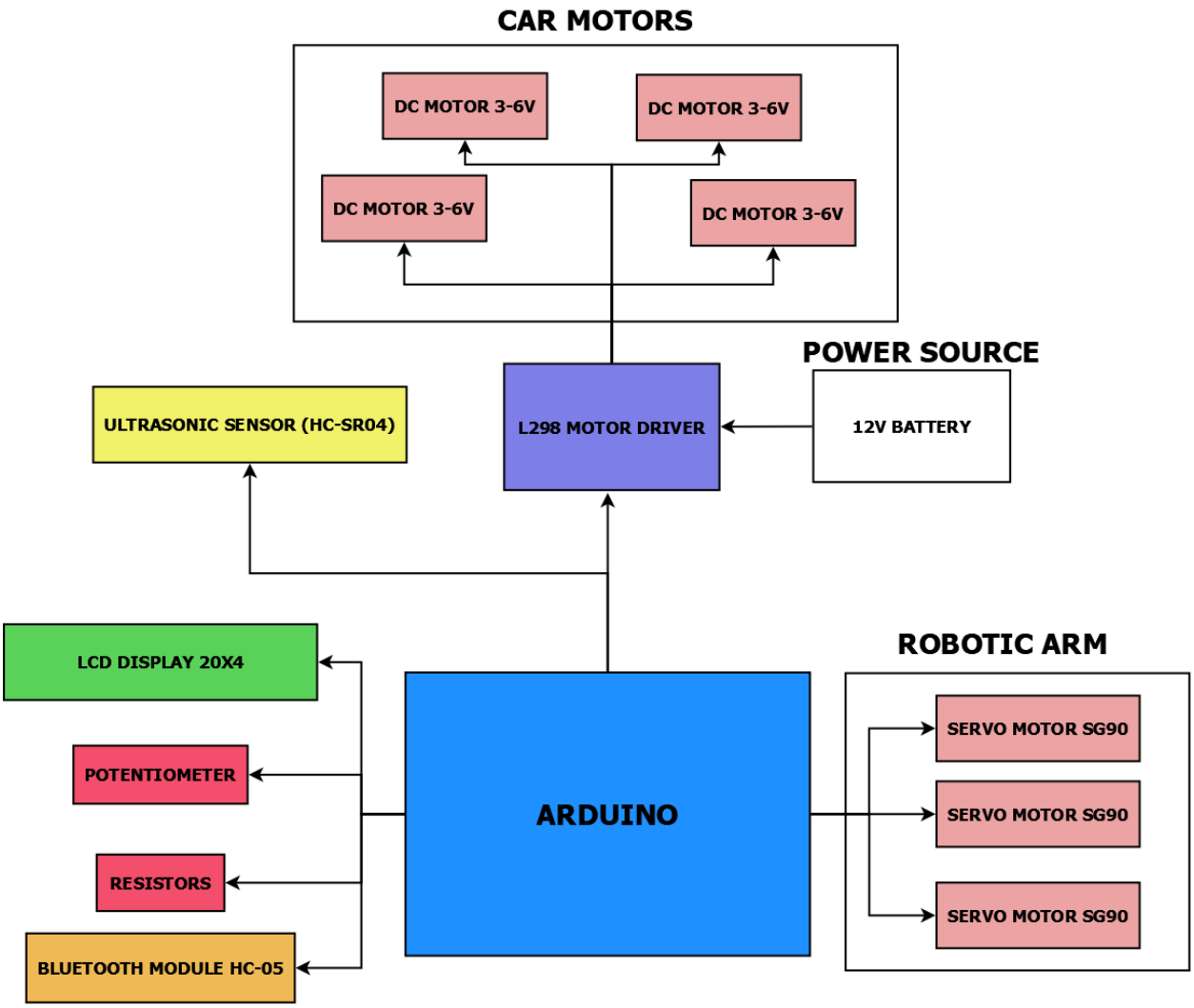
Design Criteria

We plan to use the Arduino Microcontroller to…

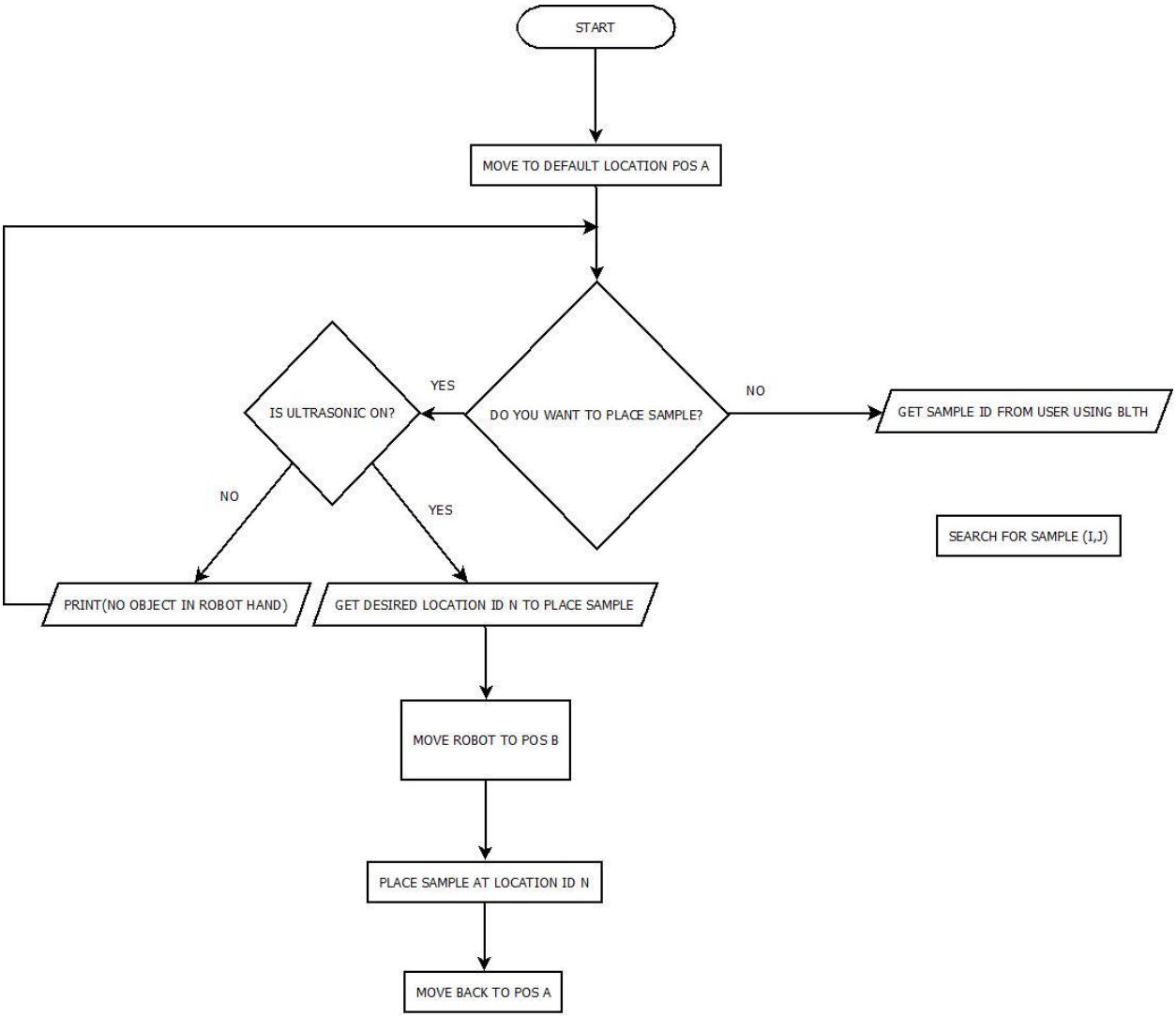
Detailed Design

Block Diagram, Circuit Layout, Additional Materials, Extracts of Pseudocode...

BLOCK DIAGRAM



FLOWCHART


START
MOVE TO DEFAULT LOCATION POS A
DO YOU WANT TO PLACE SAMPLE?
NO
GET SAMPLE ID FROM USER USING BLTH
YES
IS ULTRASONIC ON?
NO
YES
PRINT(NO OBJECT IN ROBOT HAND)
GET DESIRED LOCATION ID N TO PLACE SAMPLE
SEARCH FOR SAMPLE (I,J)
MOVE ROBOT TO POS B
PLACE SAMPLE AT LOCATION ID N
MOVE BACK TO POS A

PSEUDOCODE

1 2 3

4 5 6

```
DECLARE position_Vangle[6] SET TO [90 180 270]
DECLARE position_Hangle[6] SET TO [90 180 270]
DECLARE Position_ID[6] SET TO [1 2 3 4 5 6]
DECLARE ISITFREE[6] SET TO [-1 -1 -1 -1 -1 -1]

DECLARE direction (integer: 1 for forward, -1 for backward)
DECLARE targetSpeed SET TO 120

DECLARE ID (integer)
if start is PRESSED THEN  // if start button is PRESSED
   {
setDCmotor(motorPinA,motorPinB,-1,120)   // move the stick to the original place
setServoAngle(servoPin1,0)
setServoAngle(servoPin2,0)

//if Dr. prssed place sample and Ultrasonic detected a sample present
 if (PLACE_SAMPLE is PRESSED) AND (ULTRA_SONIC is True) THEN
    {
    count = 0  // to check the sample found empty place

        FOR i = 1 TO 6   // to loop on the positions till an empty one is found
        if (ISITFREE[i] = -1 ) THEN
            // move motors to the position of the empty slot
            setDCmotor(motorPinA,motorPinB,1,120)
            setServoAngle(servoPin1,position_Vangle[i] )
            setServoAngle(servoPin2,position_Hangle[i] )
            // state it is not free anymore
            ISITFREE[i]=1
            // quit looping
            Exit FOR
         Else
         ENDIF
         count = count +1
        NEXT i
        // check if no empty slot is found
        IF count = 5 THEN
            print ("no available places")
       ENDIF
```

```
        setDCmotor(motorPinA,motorPinB,-1,120)   // move the stick to the original place
        setServoAngle(servoPin1,0)
        setServoAngle(servoPin2,0)
       }

      if (GET_SAMPLE is PRESSED)  //IF dr requested to get a sample
         {
         count = 0 // to check if the requested sample is found
            ID <- INPUT("Enter Sample ID: ")  // enter the ID of the requested sample
              FOR i = 1 TO 6
                 if (Position_ID[i] = ID) THEN // to know the position of this ID
                    {
                       if (ISITFREE[i] = 1) THEN // to check if this position has a sample
                          {
                           // move motors to the position of the sample slot
                             setDCmotor(motorPinA,motorPinB,1,120)
                             setServoAngle(servoPin1,position_Vangle[i] )
                             setServoAngle(servoPin2,position_Hangle[i] )
                            // wait 1 sec
                             delay_in_ms(1000)
                             // return to the original place
                             setDCmotor(motorPinA,motorPinB,-1,120)   // move the stick to the
original place
                             setServoAngle(servoPin1,0)
                             setServoAngle(servoPin2,0)
                             // update the array as this position is free now
                             ISITFREE[i] = -1
                             // quit looping
                             Exit FOR
                          }
                    }
                    count = count + 1
               NEXT i
          IF count = 5 THEN // check if this sample ID is not found
              print ("WRONG ID")
         ENDIF
         }
          }

   FUNCTION setDCmotor(motorPinA, motorPinB, direction, targetSpeed)

     // Set motor direction
     IF direction == 1 THEN
       WRITE_HIGH(motorPinA)
```

```
    WRITE_LOW(motorPinB)
  ELSE
    WRITE_LOW(motorPinA)
    WRITE_HIGH(motorPinB)
  ENDIF

  // Apply PWM control for speed (similar to setDCmotorSpeed function)
  dutyCycle = map(targetSpeed, 0, maximum_speed, minimum_duty, maximum_duty)
  CONFIGURE_PWM(motorPinA, dutyCycle)  // Adjust pin depending on direction

ENDFUNCTION


FUNCTION setServoAngle(servoPin, targetAngle)

  // Convert angle to pulse width (adjust conversion based on servo specifications)
  pulseWidth = map(targetAngle, 0, maximum_angle, minimum_pulse, maximum_pulse)

  // Send pulse signal to servo pin
  WRITE_PULSE(servoPin, pulseWidth)

ENDFUNCTION
```

Forward Plan

If selected, during the ‘Build Phase’, we will spend the week as follows…

We will test the proposed design by...