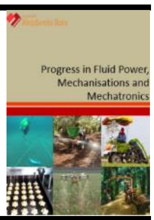
Progress in Fluid Power, Mechanisations and Mechatronics 2, Issue 1 (2018) 27-48

Penerbit Akademia Baru

Progress in Fluid Power, Mechanisations and Mechatronics

Journal homepage: www.akademiabaru.com/promecha.html
ISSN: 2672-7218



Shared Supervisory Control in Pneumatic Pick and Place Robotic System using Vision Sensor

Open Access

Ahmad Anas Yusof[1,*], Shakti Kumaran Asohkumaran[1], Muhammad Arif Aiman Ramli[1]

[1] Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

ARTICLE INFO

Article history:
Received 29 October 2018
Received in revised form 1 December 2018
Accepted 9 December 2018
Available online 10 December 2018

ABSTRACT

The motivation behind this project is to build up a vision guided pneumatic pick and place robotic system with shared supervisory control where it is anticipated that in the future, most of the operations will be conducted by autonomous robots that need visual feedback to move around the working space. In the era of internet of thing and industrial revolution 4.0, the machine has to work collaboratively with humans, to identify and locate the workpiece, to analyse the information provided by other sensors and to improve their positioning accuracy. This paper presents the use of shared supervisory control over a pneumatic pick and place cylindrical robotic system by utilizing a Pixy camera CMUcam5 as the vision sensor. The supervisory control is created by utilizing the PS2 controller as the manual mode input device that will send the command from a remote site. In a separated control, the vision sensor will act as the automatic mode input device that will guide the pick and place robot in handling two different coloured objects. Operational test results show that the shared supervisory control is well executed in both manual and automatic mode. The vision sensor also guides the pick and place robot to handle two objects in the automatic modes with successful attempts.

Keywords:
Shared supervisory control, Vision sensor, Pixy camera, Pick and place robotic system

1. Introduction

1.1 Supervisory Control

Supervisory control schemes propose the viewpoint of solving the automation problem incrementally and leaving those problems unsolved to be handled by the human supervisor. In general, the basic concept in shared supervisory control is to provide a remote-control capability for a human supervisor to generate commands to a site with the help of multiple sensors for improve situational awareness [1]. Supervisory control has been defined as an instrument that one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial effectors to the controlled process or task environment. Shared supervisory control is often in loop with two processes. Firstly,

* *Corresponding author.*
E-mail address: anas@utem.edu.my (Ahmad Anas Yusof)

the controlled machine or process continues autonomously under the observation of an operator, who manually intervenes to modify the control algorithm in certain way. Secondly, the process may accept a command and carries it out autonomously and awaits further commands when finished. Generally, in manual control, the operator interacts directly with a controlled process or task using switches and other control actuators. This concept requires a lot of human physical capabilities. By default, in automatic control, the machine adapts to shifting conditions and makes decisions in achieving some goal which can be as simple as picking up objects or switching a heating system on and off. In a broader concept, supervisory control may refer to a term for control loops within a distributed and structured control hierarchy. It refers to a Supervisory Control and Data Acquisition system or SCADA, which is used in process control, often on fairly small and remote applications such as a pipeline transport, water distribution, or wastewater utility system station. Supervisory Control and Data Acquisition (SCADA) is a control system architecture that uses computers, networked data communications and graphical user interfaces for high-level process supervisory management, but uses other peripheral devices such as programmable logic controller (PLC) and discrete PID controllers to interface with the process plant or machinery. The use of SCADA has been also considered for management and operations of project-driven-process in construction [2].

1.2 Vision Sensor

Vision-guided robots are a quickly emerging area of industrial and autonomous robotics. A vision-guided robot is basically a robot fitted with one or more cameras used as sensors to provide a primary feedback signal during automatic operation or as a secondary feedback signal to the operator for tele-operation, thus making them highly anticipated in a wide range of applications. The robots that are equipped with advanced vision systems are typically far more adept and flexible than robots that operate without a vision. Such technology is rapidly transforming production processes by enabling robots to be highly adaptable to shared supervisory control and more easily implemented [3]. The use of visual sensor specifically for grasping has resulted in numerous outcomes. One of it is the use of a calibrated coordinate transformation from a stationary camera frames to the manipulator frame. Stationary camera configuration eradicates problems with mobile cameras, but introduces the need for stereo-baseline adjustment and camera-to-manipulator coordinate transform correction. The uses of alternative to gripper, such as a basket, grapple or grabber as the end effector eliminates mechanical alignment issues, allows calibration error, and tolerates noise in both sensing and manipulator positioning. If the objects position is known than the grabber can maintain relentless actions with no speed and path changes during the manipulation process, which also simplifies the issues of camera’s calibrations, but increase system rigidity. Thus, this simplifies the robot’s mechanism, and the control system can accurately forecast the position of the object for a given point in time. There are also studies on using movable camera vision, where robot motion causes camera motion, without requiring the vision system to track the end-effector. Some studies also uses artificial neural network for object recognition in the robotic system. [4-10]. Thus this paper presents the development of a pneumatic pick and place robotic system with shared supervisory control tracks by using a vision sensor located on top of the workpieces and moves with the motion of the robots.

2. Methodology

This section describes the programming and test rig setup for the test. The functionality of the shared supervisory control is one of the crucial aspects for the pneumatic pick and place robotic system to enhance the performance and productivity of the system. An operational test has been conducted based on manual and automatic mode operations. A wireless PS2 controller with its transmitter and receiver is used to control the pick and place robotic system. The control system has been developed by using Arduino Uno microcontroller and the SKPSW wireless transmitter. The related button on the PS2 controller is used to switch a pre-programmed manual and automatic movement based on C programming language. The gripper releases the grasp when the push button X is released. Downward movement is controlled by button L1, while R1 controls the forward movement. Gripper and the whole robot rotation movement are controlled by button □ and ▶, respectively.

Table 1
Assigned connection for relays

Function	Button
Gripper Operation	X
Downward Movement	L1
Forward Movement	R1
Gripper Rotation	□
Robot Rotation	▶

2.1 Programming Setup

In this section, the algorithm and the pseudocodes for the shared supervisory control are presented. The controller is based on Arduino Uno microcontroller, with Arduino IDE software, which allows researcher to write sketches, a form of Arduino programming language. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. Note that the programming is suited for manual and automatic mode, using five 5/2-way pneumatic directional control valves, which controls five degree of movement of the pick and place robotic system. Full flow chart and Arduino sketches can be observed in Appendix A and B.

2.1.1 Algorithm

The algorithm presents in this project defines the initialization and operation set of steps for the test. The initialization algorithm involves the setting parameters, the outputs and the inputs in the program. The written algorithm for the operation test is as follows:

```
Program to run Pick and Place System using Microcontroller


Initialize

        Set    Predetermine location
               Manual Mode
               Automatic Mode
               Pixy Signature 1 colour
```

```
                Pixy Signature 2 colour
                Pixy Camera
                PS2 Shield
        Output
                Gripper
                Horizontal Cylinder
                Vertical Cylinder
                Gripper Rotary Actuator
                Main Rotary Actuator
        Input
                Pixy Signature 1
                Pixy Signature 2
                PS2 Button X
                PS2 Button L1
                PS2 Button R1
                PS2 Button R2
                PS2 Button □
                PS2 Button ▶

Operation


        If PS2 Button R2 pushed in Automatic Mode
                Change to Manual Mode
        If PS2 Button R2 pushed in Manual Mode
                Change to Automatic Mode
        If PS2 Button X pushed in Manual Mode
                Gripper Grasps the Object
        If PS2 Button X released in Manual Mode
                Gripper Ungrasps the Object
        If PS2 Button R1 pushed in Manual Mode
                Horizontal Cylinder Extends

        If PS2 Button R1 released in Manual Mode
                Horizontal Cylinder Retracts
        If PS2 Button L1 pushed in Manual Mode
                Vertical Cylinder Extends
        If PS2 Button L1 released in Manual Mode
                Vertical Cylinder Retracts
        If PS2 Button □ pushed in Manual Mode
                Gripper Rotary Actuator Rotates Counter Clockwise
        If PS2 Button □ released in Manual Mode
                Gripper Rotary Actuator Rotates Clockwise
        If PS2 Button ▶ pushed in Manual Mode
                Main Rotary Actuator Rotates Counter Clockwise
        If PS2 Button ▶ released in Manual Mode
                Main Rotary Actuator Rotates Clockwise

        If Pixy Signature 1 in Automatic Mode at predetermined location
                Vertical Cylinder Extends
                Gripper Grasps the Object
                Vertical Cylinder Retracts
                Horizontal Cylinder Extends
                Vertical Cylinder Extend
                Gripper Ungrasps the Object
                Vertical Cylinder Retracts
                Horizontal Cylinder Retracts

        If Pixy Signature 2 in Automatic Mode at predetermined location
                Vertical Cylinder Extends
                Gripper Grasps the Object
                Vertical Cylinder Retracts
                Main Rotary Actuator Rotates Counter Clockwise
                Gripper Ungrasps the Object
                Main Rotary Actuator Rotates Clockwise
```

2.1.2 Pseudocodes

In this research, the pseudocodes is written in a detailed yet readable description of what a shared supervisory control of a pick and place robotic system must do, and it is expressed in a formally-styled natural language rather than in a programming language. The pseudocodes for the experiment use high level style syntax such as IF, Else-IF and Call, which describes the selections in the program. The written pseudocodes for the test are as follows:

```
function Automatic Mode Signature 1
       {
              Automatic Mode Signature 1;
              Vertical Cylinder Extends;
              Gripper Grasps the Object;
              Vertical Cylinder Retracts;
              Horizontal Cylinder Extends;
              Vertical Cylinder Extends;
              Gripper Ungrasps the Object;
              Vertical Cylinder Retracts;
              Horizontal Cylinder Retracts;
              Return;
              }
function Automatic Mode Signature 2
              {
              Automatic Mode Signature 2;
              Vertical Cylinder Extends;
              Gripper Grasps the Object;
              Vertical Cylinder Retracts;
              Main Rotary Actuator Rotates Counter Clockwise;
              Gripper Ungrasps the Object;
              Main Rotary Actuator Rotates Clockwise;
              Return;
              }
function Manual Mode
              {
              Manual Mode;
              if (X button press)
              {
              Gripper Grasps the Object;
              }
              else
              {
              Gripper Ungrasps the Object;
              }
              if (R1 button press)
              {
              Horizontal Cylinder Extends;
              }
              else
              {
              Horizontal Cylinder Retracts;
              }
              if (L1 button press)
              {
              Vertical Cylinder Extends;
              }
              else
              {
              Vertical Cylinder Retracts;
              }
              if ([] button press)
              {
              Gripper Rotary Actuator Rotates Counter Clockwise;
```

```
}
else
{
Gripper Rotary Actuator Rotates Clockwise;
}
if (> button press)
{
Main Rotary Actuator Rotates Counter Clockwise;
}
else
{Main Rotary Actuator Rotates Clockwise;
}
Return;
}

Start;
Sync transmitter and receiver of SKSPW;
if(R2 button press)
{{{
Automatic Mode;
Pixy Camera Scan;
if (Pixy Camera Detects Signature 1)
{
call Automatic Mode Signature 1;
}
Pixy Camera Scan;
if   (Pixy Camera Detects Signature 2)  {
call Automatic Mode Signature 2;
}
  Pixy Camera Scan;
  Stop;
  return;
}}}
Manual Mode;
call Manual Mode;
Stop;
```

2.2 Test Rig Setup

In the test, the pick and place robotic system with five pneumatic cylinders are used for the robot and gripper rotation, downward and forward movement, and for the grasping process [10]. The pneumatic pick and place robotic system is pre-programmed for a shared supervisory control, using Arduino Uno microcontroller that has been programmed for manual and automatic control. The PS2 wireless Joypad is used to control the movements manually, while Pixy camera CMUcam5 serves as the vision sensor that processes images captured into the automatic pre-programmed movement. [11-12]. Figure 2 illustrates the Pixy camera used in the test while Fig. 3 shows the assigned movement of pneumatic pick and place system in manual mode. Relays are used to control the solenoid valves, which eventually control the pneumatic system. Prior to the pneumatic pick and place operation, the Pixy camera is positioned and calibrated on top of the pneumatic gripper, facing down directly on a pre-designated position located under the gripper. This is important due to the end-to-end movement of the system, since the pneumatic system depends on the 5/2- way directional control valves. The angle of Pixy facing downward is also crucial in order for the Pixy to detect objects with suitable distance under the gripper for automatic mode operation. Fig. 4 illustrates the detection of orange and red objects during the automatic mode operation test.

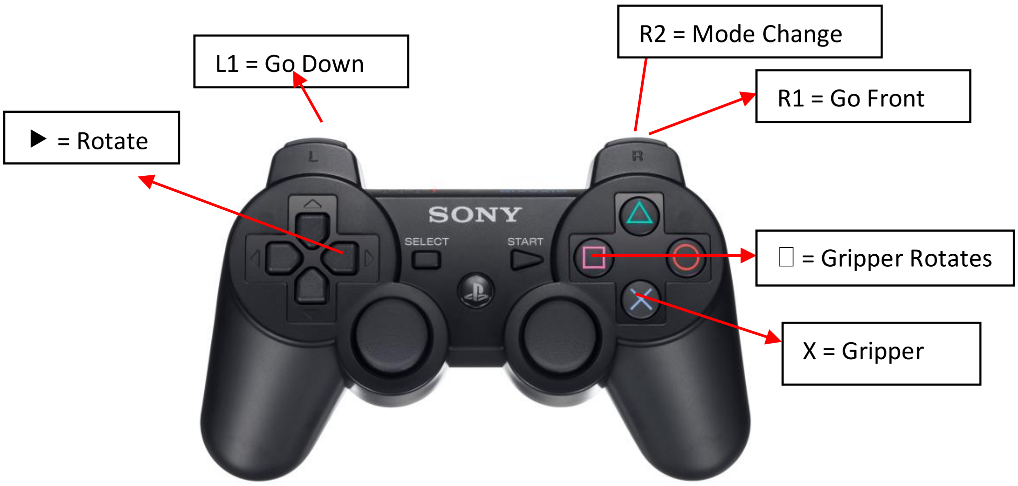


Fig. 1. PS2 joypad controller with assigned function

Fig. 2. pixy camera CMUcam5

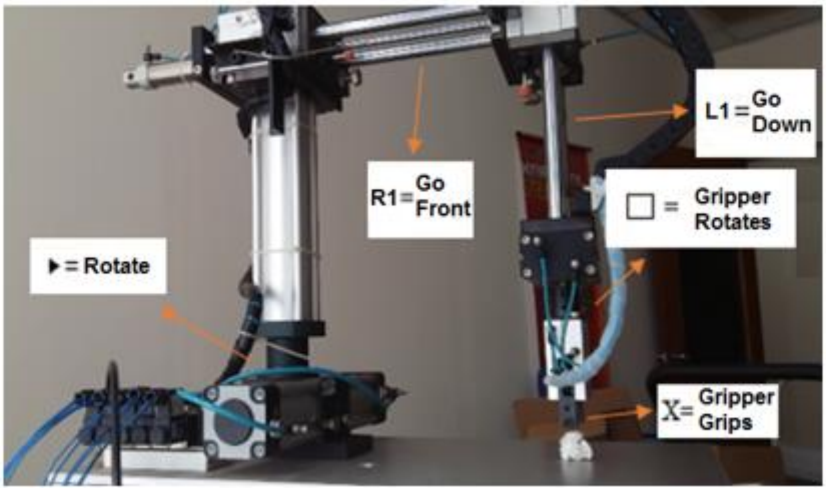


Fig. 3. pneumatic pick and place robotic system with assigned movements

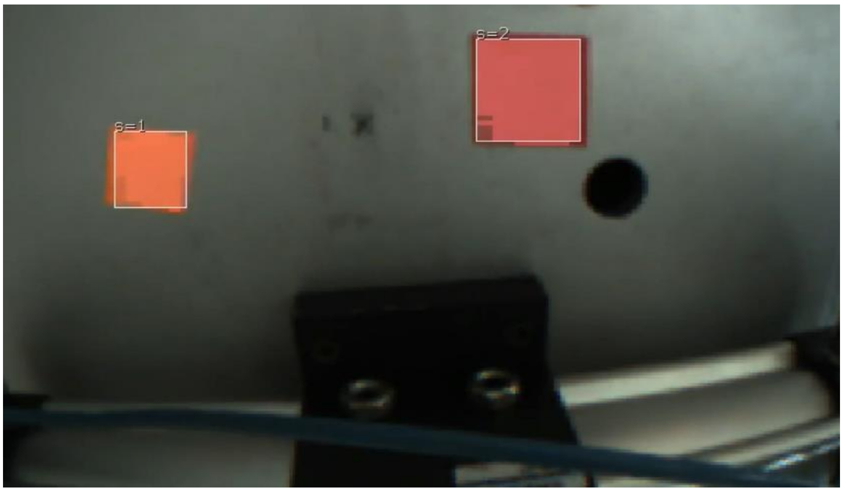


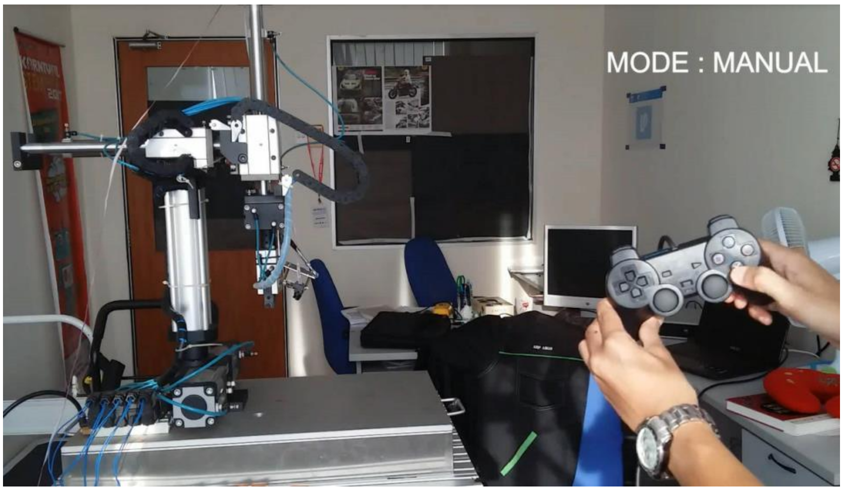
Fig. 4. Detection of orange and red objects for automatic movements

3. Results and Discussions

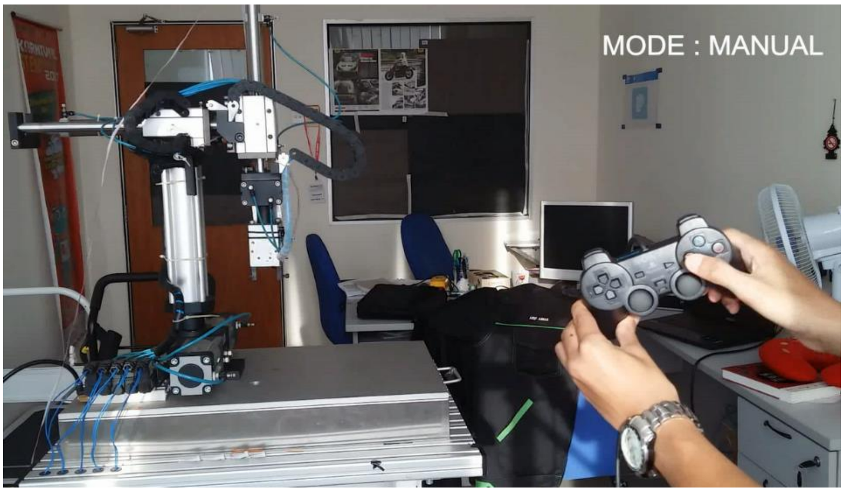
The test results on the communication between the Pixy camera, the PS2 controller and the pneumatic pick and place robotic system have been illustrated in Fig. 5 to Fig. 7. The operation test has been conducted in order to perform the required task in manual and automatic mode and to check for the functionality of the vision sensor and the shared supervisory control.

3.1 Manual Mode Operational Test

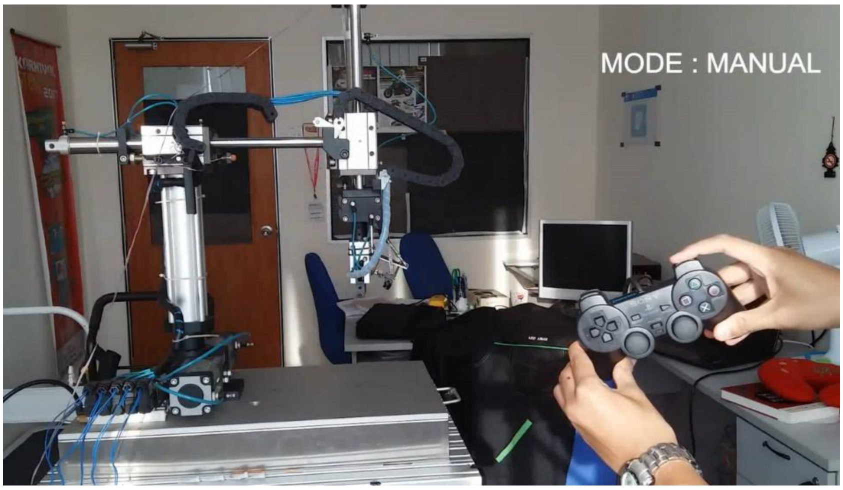
In manual mode control, five buttons are used to control five movements of the pneumatic pick and place robotic system. The operator may use the shared supervisory control by pushing button R2, to interchangeably opt for manual or automatic mode. Fig. 5(a) shows that the gripper grasping process when button X is pushed. All movements are illustrated in Fig. 5(a) to Fig. 5(e).



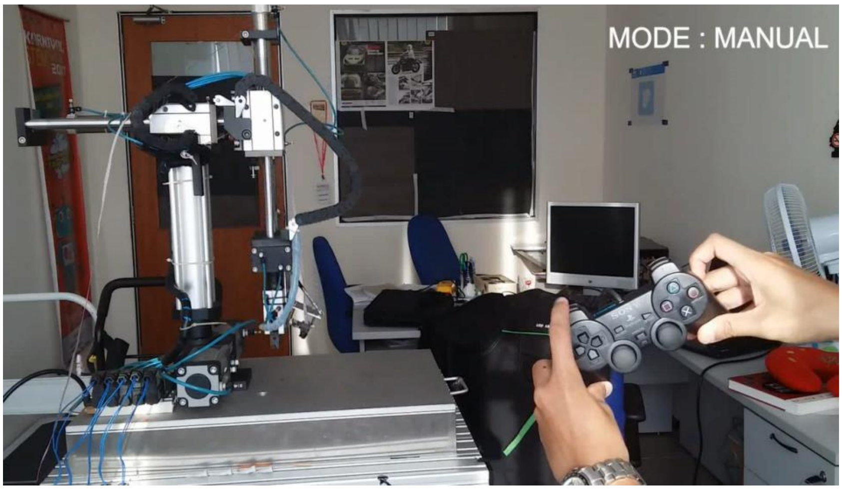
(a) gripper grasps the object by pushing button X



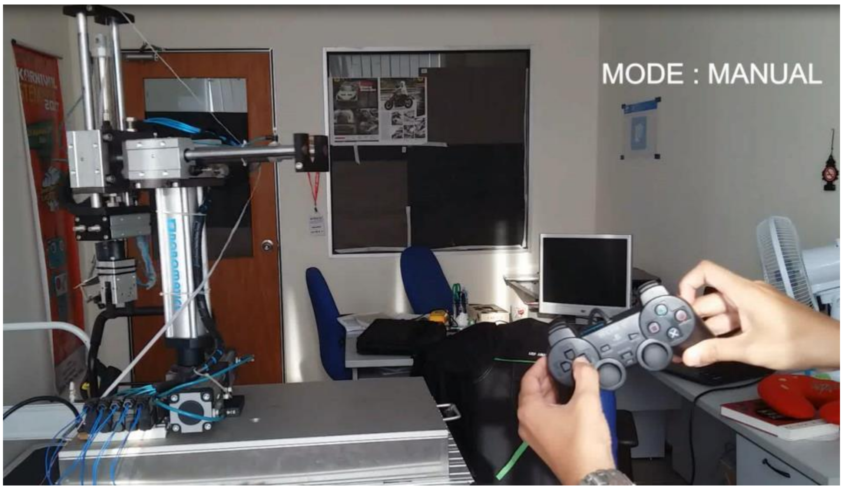
(b) gripper rotates by pushing button □



(c) Pick and place robot extend forward by pushing button R1



(d) pick and place robot extend downward by pushing button L1

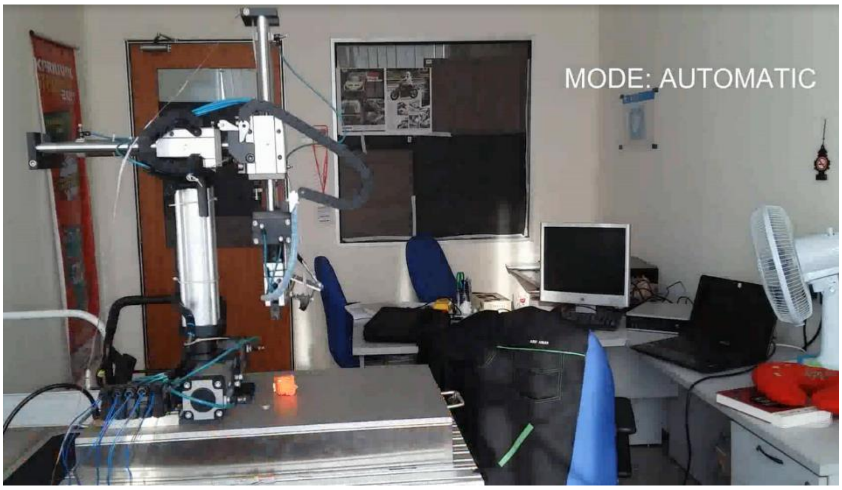


(e) pick and place robot rotates 180^{o} anti-clockwise by pushing button ▶

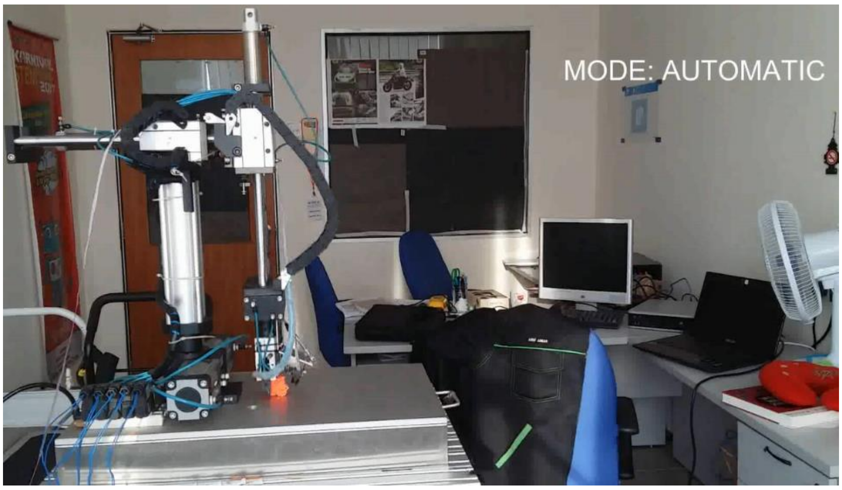
Fig. 5. Robot's movement in manual mode.

3.2 Automatic Mode Operational Test

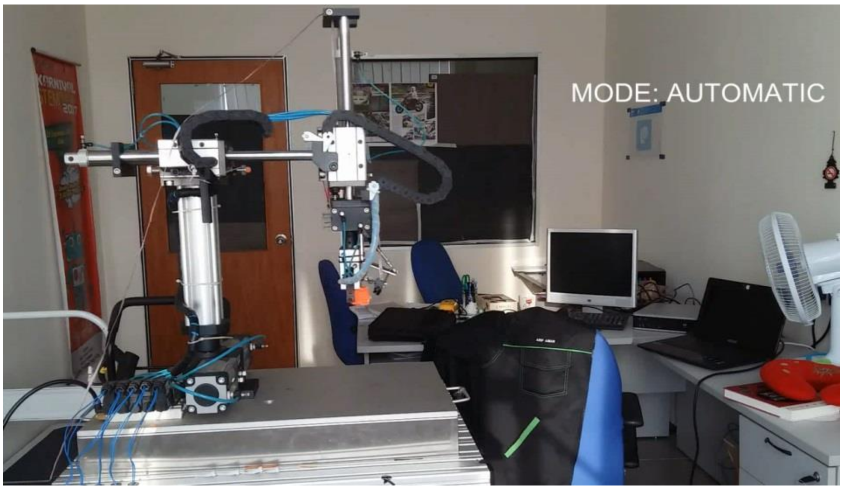
In the automatic mode operational test, the pick and place robotic system is subjected to a simple material handling routine with two objects of different colours. The robot is pre-programmed to detect red and orange objects. The pixy camera will then detect the object in a predetermined location of the object. If the orange object is detected, the pick and place robotic system will move vertically downward at a fixed speed of 4 cm/s along the y-axis of the image, as illustrated in Fig. 6(a) and Fig. 6(b). The gripper then grasps the object, and moves vertically upward with the same velocity. Upon reaching the maximum height, the arm then moves forward to convey the object to a new location. The location of the target is designated at the end of the fully extended condition. At the designated target, the robot will then move downward, and release the object as illustrated in Fig. 6(c) to Fig. 6(e).



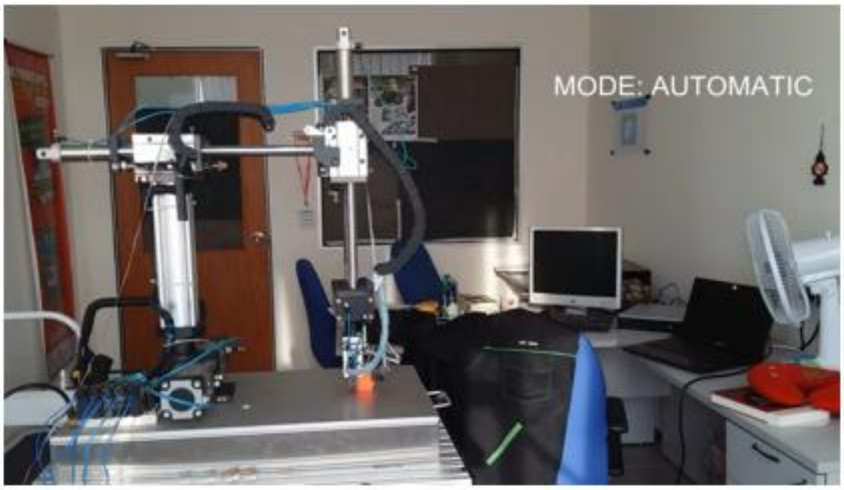
(a) gripper moves downward to grasp the orange object at the designated starting point



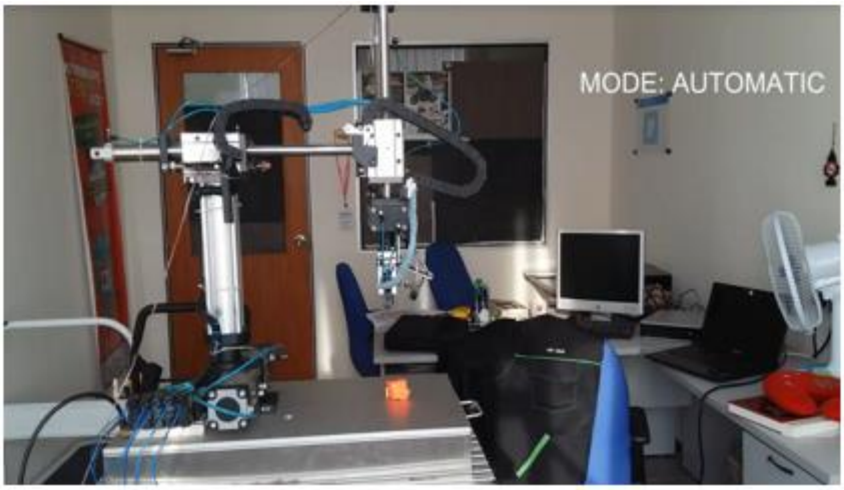
(b) gripper grasps the orange object



(c) object is raised, transferred horizontally to a designated receiving point



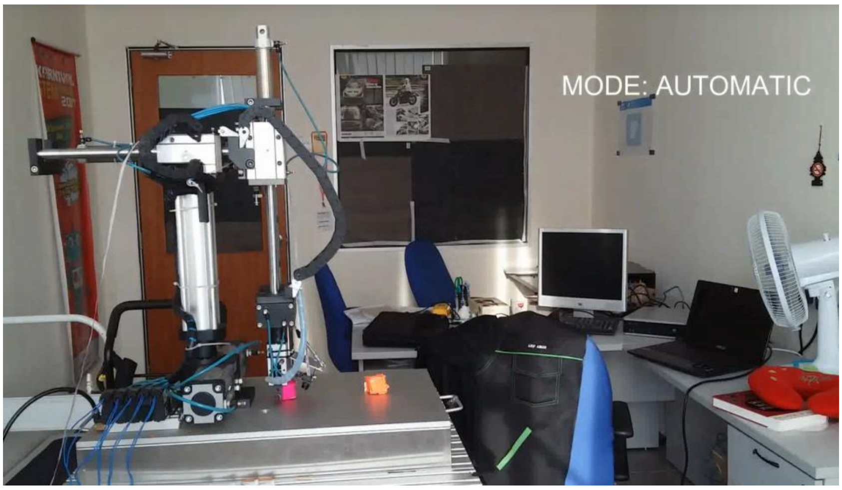
(d) gripper transfers the object to the designated position



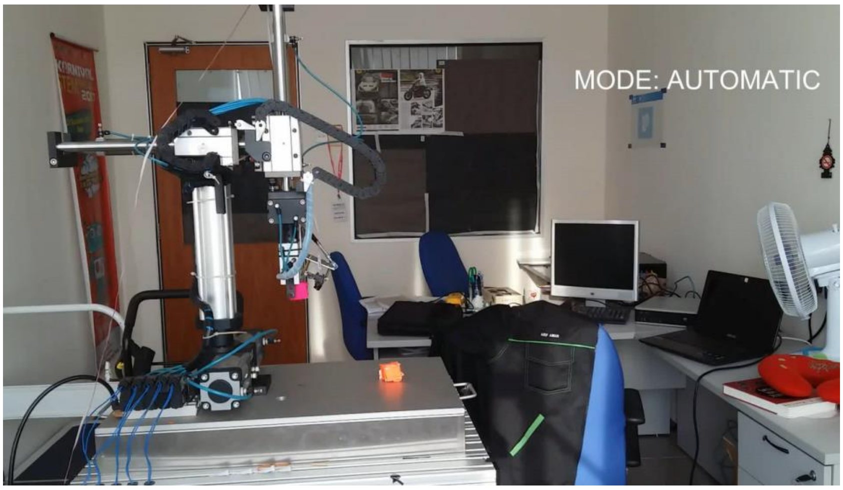
(e) gripper releases the object

Fig. 6. Grasping process of an orange object

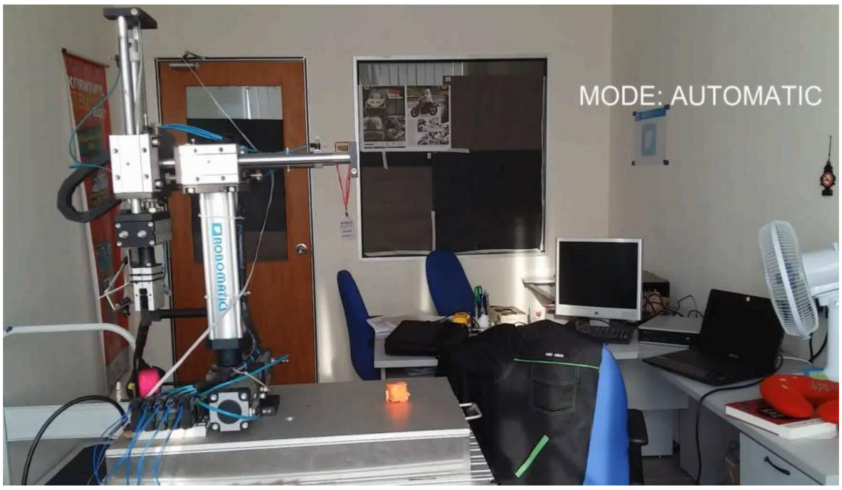
The object will be continuously scanned and captured and analyzed by the microcontroller and the pixy camera. If a red object is detected, the pick and place robot will follow the same step, based on the semi-cylindrical coordinate movement system. The gripper will move vertically downward at a fixed speed of 4 cm/s along the y-axis of the image, as illustrated in Fig. 7(a) and Fig. 7(b). The gripper then grasps the object, and moves vertically upward with the same velocity. Upon reaching the maximum height, the arm then rotates to the left in an 180^{o} counter-clockwise swing to convey the object to a new location. The location of the target is designated at the end of the fully rotation condition. At the designated target, the robot will then release the object as illustrated in Fig. 7(c).



(a) gripper grasps the red object at the designated starting point



(b) gripper grasps and lifts the red object



(c) gripper transfer the object to another designated position and release it

Fig. 7. Grasping process of a red object

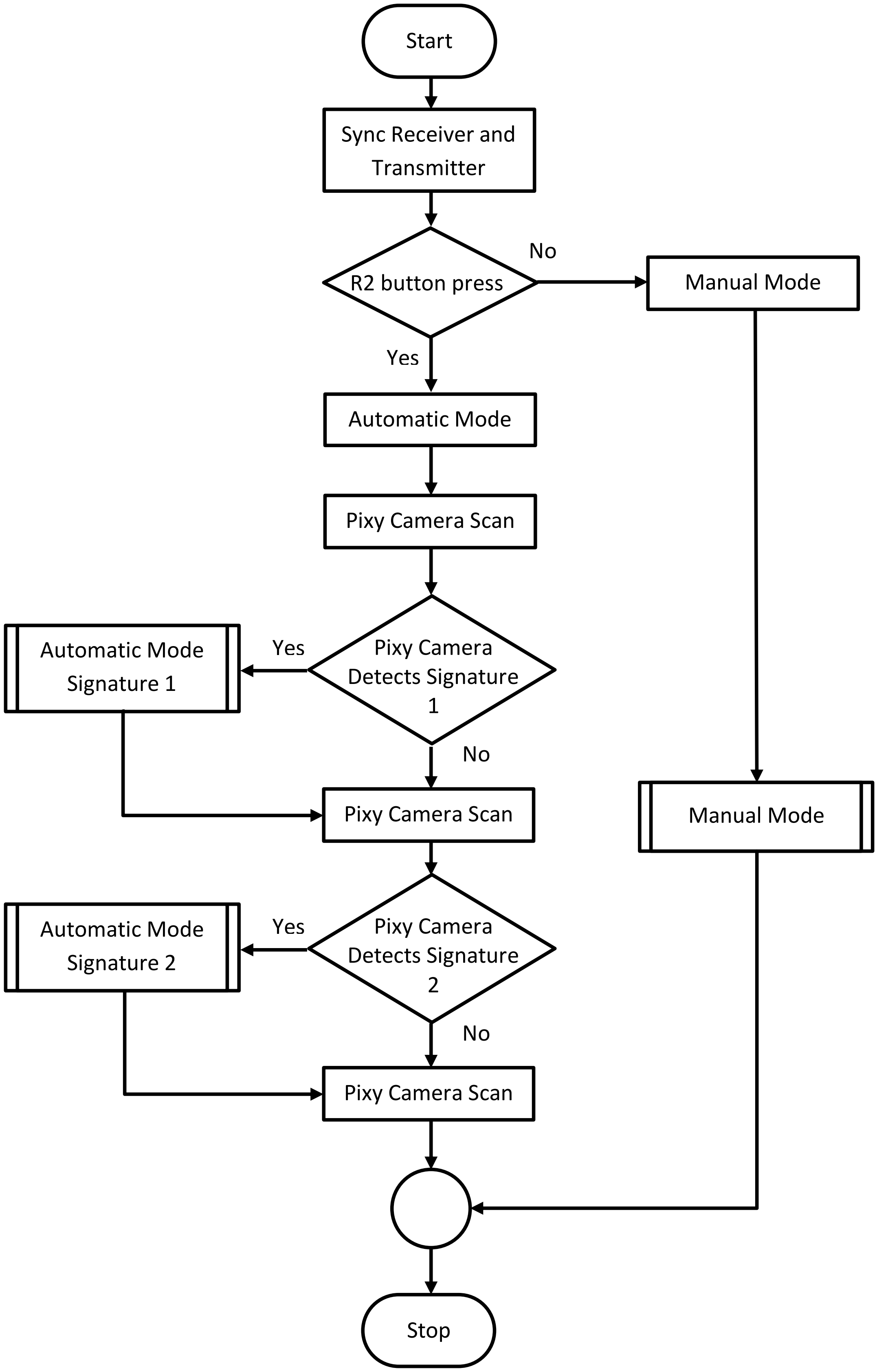
4. Conclusions

The study on the development of the pneumatic pick and place robotic system with shared supervisory control has been presented in this paper. There are many ways of control method in the development of a pick and place robotic system. The Arduino microcontroller has been used in the study, with program codes that is introduced to operate the pneumatic pick and place robotic system in manual and automatic mode, for shared supervisory control. Pixy camera CMUcam5 is used as the main input sensor to work as the vision sensor during the automatic mode operation test. The test has been conducted in order to understand the ability of the Pixy camera to detect and differentiate red and orange in different circumstances. It is noted that the pick and place robotic system manages to differentiate between both coloured objects, and transferred the objects to the designated locations. However, due to the usage of 5/2-way directional control valves on the system, it can only operate in the end-to-end movement, which limits the object to be placed only at a specific designated target and not at any random positions. Future development will involve the improvement of the system in handling objects at random positions.

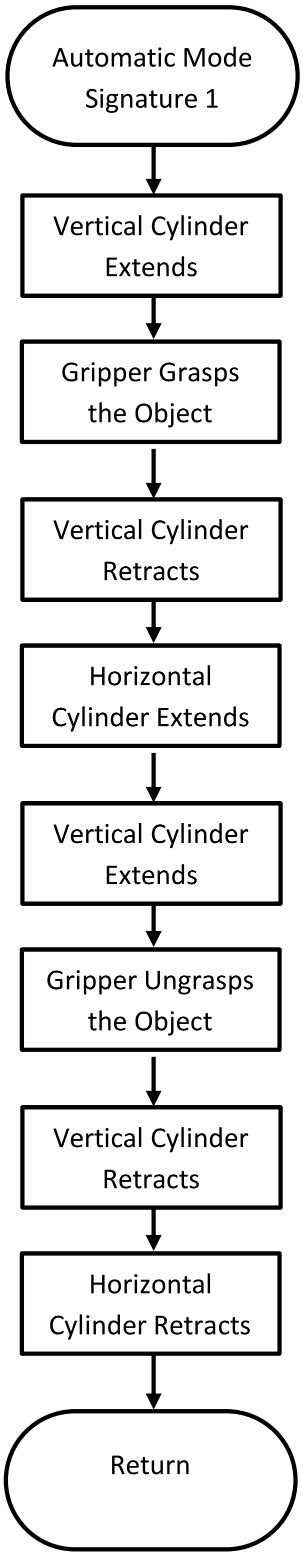
References

[1] Salvendy, Gavriel, ed. *Handbook of human factors and ergonomics*. John Wiley & Sons, 2012.
[2] Lee, Lian-Wang, Hsin-Han Chiang, and I. Li. "Development and Control of a Pneumatic-Actuator 3-DOF Translational Parallel Manipulator with Robot Vision." *Sensors* 19, no. 6 (2019): 1459.
[3] Allen, Peter K., Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. "Automated tracking and grasping of a moving object with a robotic hand-eye system." *IEEE Transactions on Robotics and Automation* 9, no. 2 (1993): 152-165.
[4] Brandt, Scott A., Christopher E. Smith, and Nikolaos P. Papanikolopoulos. "The Minnesota Robotic Visual Tracker: a flexible testbed for vision-guided robotic research." In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1363-1368. IEEE, 1994.
[5] Buttazzo, Giorgio C., Benedetto Allotta, and Felice P. Fanizza. "Mousebuster: a robot system for catching fast moving objects by vision." In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pp. 932-937. IEEE, 1993.
[6] Houshangi, Nasser. "Control of a robotic manipulator to grasp a moving target using vision." In *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 604-609. IEEE, 1990.
[7] Koivo, A. J. "On adaptive vision feedback control of robotic manipulators." In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pp. 1883-1888. IEEE, 1991.
[8] Schrott, Anneliese. "Feature-based camera-guided grasping by an eye-in-hand robot." In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 1832-1837. IEEE, 1992.
[9] Charmedlabs, *Pixy CMUcam5*. http://charmedlabs.com/default/pixy-cmucam5/ Accessed April 1, 2018.
[10] Yusof, Ahmad Anas, Ihsan Ismail, Syarizal Bakri, Mohd Khairi Mohamed Nor, and Rabiah Ahmad. "Preparation, Field Observation and Survey on STEM Engagement of 5-Axis Pick-and-Place Pneumatic Robotic Arm Remote-Manipulation." In *Intelligent Manufacturing & Mechatronics*, pp. 219-230. Springer, Singapore, 2018.
[11] Candelas, F. A., Giggs J. García, Santiago Puente, J. Pomares, C. A. Jara, J. Pérez, D. Mira, and F. Torres. "Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics." *IFAC-PapersOnLine* 48, no. 29 (2015): 105-110.
[12] Arduino. *Arduino Board Uno*. https://store.arduino.cc/usa/arduino-uno-rev3. Accessed April 1,2018.

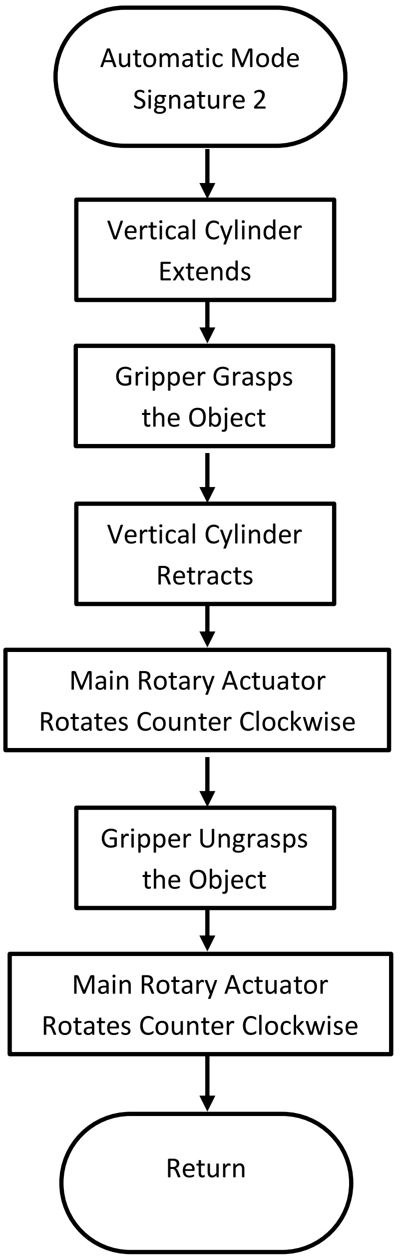
Appendix A : Main Program Flow Chart



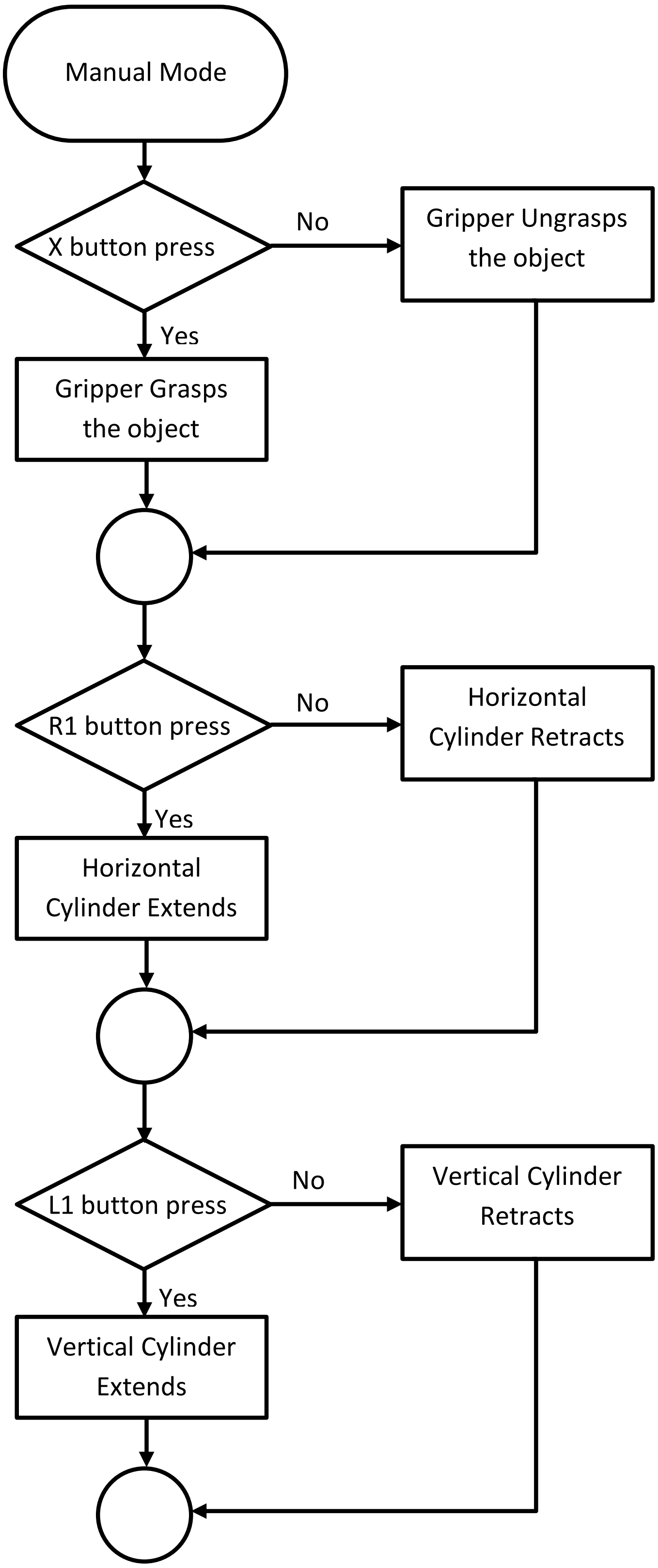
Appendix B : Automatic Mode Signature 1 Function Flow Chart



Appendix C : Automatic Mode Signature 2 Function Flow Chart

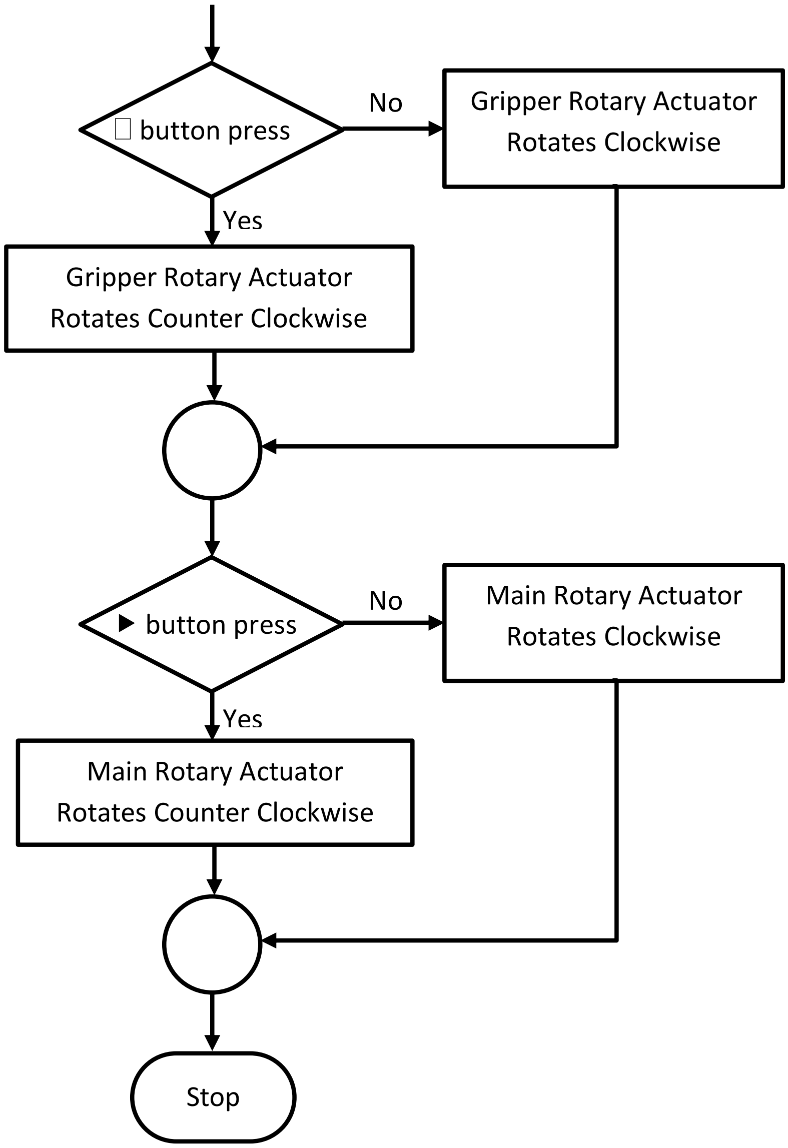


Appendix D : Manual Mode Function Flow Chart



Progress in Fluid Power, Mechanisations and Mechatronics
Volume 2, Issue 1 (2018) 27-48

Appendix D : Manual Mode Function Flow Chart (continue..)



Appendix E : Arduino Sketches

```
// Writing Arduino sketches for controlling five 5/2-way valves on a
// pneumatic pick and place system with shared supervisory control in the
// form of manual mode and automatic mode operation. The automatic mode
// detects 2 objects with different colours in pre-determined end-to end
// position.
// Codes written by Muhammad Arif Aiman bin Ramli B041410233
// 11 April 2018
// Faculty of mechanical Engineering
// Universiti Teknikal Malaysia Melaka

#include <SoftwareSerial.h> //declare class object
#include <Cytron_PS2Shield.h>      //PS2 ps2 = PS2(); //PS2 class object: ps2
#include <SPI.h>
#include <Pixy.h>
Cytron_PS2Shield ps2(2,3);
Pixy pixy;
int signature = 0;
int x = 0; //positon x axis
int y = 0; //position y axis
int Xmin = 70; //min x position
int Xmax = 200; //max x position
int Ymin = 60; //min y position
int Ymax = 140; //max y position
static int i = 0;
int c=2;
#define gripper 6
#define front 9
#define head 8
#define down 10
#define turn 7
void setup()
{
pinMode(front,OUTPUT);
pinMode(gripper,OUTPUT);
pinMode(head,OUTPUT);
pinMode(down,OUTPUT);
pinMode(turn,OUTPUT);
Serial.begin(9600);
ps2.begin(9600);  //initialize  the  main  board  to  use  desired  (baudrate,  rx,
tx)//This baudrate must same with the jumper setting at PS2 shield
Stop();
pixy.init();
}
void loop()
{
while(millis()<5000)
{
scan();
}
uint16_t blocks;
blocks = pixy.getBlocks();
scan();
if(c==1)
{
if(ps2.readButton(PS2_RIGHT_2) == 0)
//      if button R2 is pressed (0 = pressed, 1 = released)
{
c=2;
}
else if(signature==1 && blocks && x>=Xmin && x<=Xmax && y>=Ymin && y<=Ymax)
{
sig1();
}
```

```
else if(signature==2 && blocks && x>=Xmin && x<=Xmax && y>=Ymin && y<=Ymax)
{
sig2();
}
else
{
Stop();
}
}
else if(c==2)
{
if(ps2.readButton(PS2_RIGHT_2) == 0) // if button R2 is pressed (0 = pressed, 1
= released)
{
c=1;
}
if ( ps2.readButton(PS2_SQUARE) == 0) //0 = pressed, 1 = released
{
digitalWrite(head, HIGH);
}
else
{
digitalWrite(head, LOW);
}
if ( ps2.readButton(PS2_CROSS) == 0)
{
digitalWrite(gripper, HIGH);
}
else
{
digitalWrite(gripper, LOW);
}
if ( ps2.readButton(PS2_RIGHT) == 0)
{
digitalWrite(turn, HIGH);
}
else
{
digitalWrite(turn, LOW);
}
if ( ps2.readButton(PS2_LEFT_1) == 0)
{
digitalWrite(down, HIGH);
}
else
{
digitalWrite(down, LOW);
}
if ( ps2.readButton(PS2_RIGHT_1) == 0)
{
digitalWrite(front, HIGH);
}
else
{
digitalWrite(front, LOW);
}
}
}
void sig1() // when signature 1 detected, run this function {
digitalWrite(head,LOW);
digitalWrite(turn,LOW);
digitalWrite(down,LOW);
digitalWrite(front,LOW);
delay(100);
digitalWrite(gripper,HIGH);
delay(700);
digitalWrite(gripper,LOW);
delay(700);
```

```
digitalWrite(down,HIGH);
delay(2000);
digitalWrite(gripper,HIGH);
delay(1000);
digitalWrite(down,LOW);
delay(2000);
digitalWrite(front,HIGH);
delay(3000);
digitalWrite(down,HIGH);
delay(2000);
digitalWrite(gripper,LOW);
delay(1000);
digitalWrite(down,LOW);
delay(2000);
digitalWrite(front,LOW);
delay(3000);
}
void sig2() // when signature 2 detected, run this function {
digitalWrite(head,LOW);
digitalWrite(turn,LOW);
digitalWrite(down,LOW);
digitalWrite(front,LOW);
delay(100);
digitalWrite(gripper,HIGH);
delay(700);
digitalWrite(gripper,LOW);
delay(300);
digitalWrite(down,HIGH);
delay(2000);
digitalWrite(gripper,HIGH);
delay(1000);
digitalWrite(down,LOW);
delay(2000);
digitalWrite(turn,HIGH);
delay(3000);
digitalWrite(gripper,LOW);
delay(1000);
digitalWrite(turn,LOW);
delay(3000);
}
void Stop() // stop the running program then set the system to the default position
{
digitalWrite(front,LOW);
digitalWrite(gripper,LOW);
digitalWrite(head,LOW);
digitalWrite(down,LOW);
digitalWrite(turn,LOW);
}
void scan() // scan the object via Pixy
{
uint16_t blocks;
blocks = pixy.getBlocks(); //receive data from pixy
signature = pixy.blocks[i].signature;     //get object's signature
x = pixy.blocks[i].x; //get x position y = pixy.blocks[i].y; //get y position }
```