

**Robotics**

**Yongjie Yang**

**09/18/2020**

**Boston University**

**Electrical and Computer Engineering Department**

I. Introduction

1. Background

Nowadays, with the continuous innovation of computer hardware and software technology, artificial intelligence develops rapidly. As a result, the robotics technology we once dreamed of is constantly improving and optimizing. Especially at present, most factories use robotics to gradually replace the simple and repetitive works, so that works can free their hands and government can release labor force. Therefore, how to enable the robotics to quickly learn and master a corresponding factory technology is my current research topic.

1. Product Statement

The robot is designed to recognize and calculate results by its core chip through different inputs, for example: camera, depth sensors, shape sensors and graphics sensor, finally, and the results from core chip can transmit to the robotic arm. To be specific, for example, there is a box on its way to the end of a conveyor belt. At the same time, the inputs of the robot, such as camera and graphics sensors, can recognize the shape and size of the box.

II. Applications

1. Introduction

In the following process, I will mainly describe two main applications I have

collected from open source. One is Armrobot, and another one is Ball Tracking Robot.

1. Armrobot

Link: <https://github.com/ftobler/robotArm>

License: N/A

Languages: C++, Arduino

The first one is widely used in normal studies, which can meet most of the instructions and contents set by us. Its name is ArmRobot. Moreover, this robot has certain flexibility. Due to its size and the simplicity of the motor, it cannot complete larger objects and more complex instructions. In additional, from the figure 1. It is not difficult to find that this mechanical robot has only two pullets that could be turned. However, I find that the pulleys on the head can only grasp objects in four directions: up, down, left and right, instead of the 360-degree rotation. Especially in the coding files, as for the problems existing in the control system, I am confused that find out how its robot locates the origin (0,0,0) in the open source. Although on a normal conveyor belt, we just use the robot to grab what we need, and it doesn’t matter which plane of the staff we grab. However, if only one side of the product is smooth and easy to grasp, our robot should know the shape and size of the object in advance. To be specific, the robot should know its capacity to catch the objects rather than keep trying to grab it all the time. Therefore, my suggestion is to add cameras and other sensors to the original basic conditions so that the robot can self-perceive the shape and size of the target object and it can realize the command through its own judgment. For every robot, the motor is often the most important part, and how to control the motor is equally important, because the decision to pick a motor tends to better determine the ArmRobot’s ability to perform the operation. For example, we may use milliseconds or microseconds for time delays in the Arduino or C++. Since instead of recording the reaction time of our microcontroller, the reaction time of the motor has already occupied a large part. In this project, because the robot was small, it chose an inexpensive motor that could be seem daily, and time delays in the code were often measured in milliseconds or seconds. It is not criticized that this way is bad, because it is noticed that it is important to incorporate the actual motor in the pre-code process to create better results.

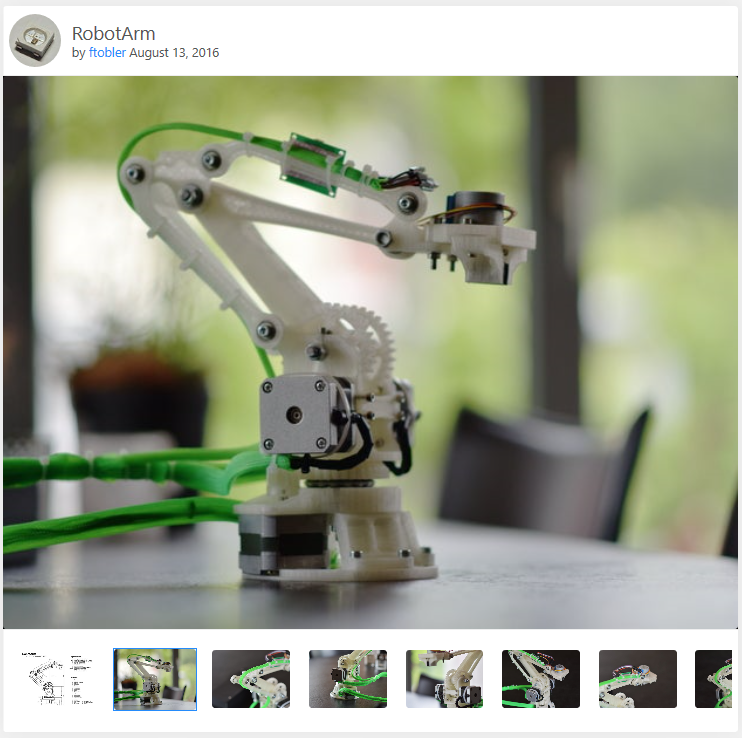


Figure 1. 3D print ArmRobot

To be honest, this ArmRobot has many shortcomings not only in the mechanical design class, but it also has problems in the code and math. However, this author has been working hard to modify and ensure the hard update, which is reason why I chose this project. Besides, for those stuff that could be modified, I will talk about them in the next application. Compared with other 3D printing robot arm applications, this product is more flexible so that it would have a better play space. Actually, I have found a near-perfect product robot, but unfortunately, the contributors are inactive almost one year long.

1. Ball Tracking Robot

Link: <https://create.arduino.cc/projecthub/junejarohan/ball-tracking-robot-7a9865?ref=user&ref_id=46171&offset=0>

License: *Copyright (C) 2019-2020, Xperience AI, all rights reserved.*

Language: python by Raspberry Pi 2 Model B, Arduino

I find a very interesting project about camera sensing Ball Tracking Robot. Although the function of robot is not to record data on the conveyor belt, its practical use could be widely used. In a nutshell, this Ball Tracking Robot is a small car robot equipped with sensors to autonomously monitor the area. It is interesting to point that the car tracks ball not limiting in the punctuation of a two-dimensional map. In other words, a 2D map with X-axis and Y-axis is mathematically positioned first in its computer, which is Raspberry Pi. Besides, it also sets the color, shape and size of the target object, which is exactly what our robot needs. One of the great things that it has a 360-degree composition to depict the object. In addition, in this project, the most intelligent point is the motor controlling. The author adopts a mechanical structure called H-bridge to control the motor of the car.

Based on this project, I have a bold idea. I am going to try to integrate two aspects in implementation. Applying the working principle of ArmRobot, and adding camera and other induction functions on the basis, it could be successfully recognizing a target for my robot and responding to commends.

III. Implementation

I will try to design a robot that could meet the following functions at the code level. Firstly, the robot can read the inputs transmitting messages. Then the commends will be computed by a microcontroller. The processed data is fed back to the arm of the robot through digital signals. Because this is a really linear process, I will not give any storage module to temporally store any data from our inputs or our results of the calculation. In figure 2, I provide a simple flow chart architecture. I have not yet decided whether to use two development boards for this project. And I will be updating my progress on Github.

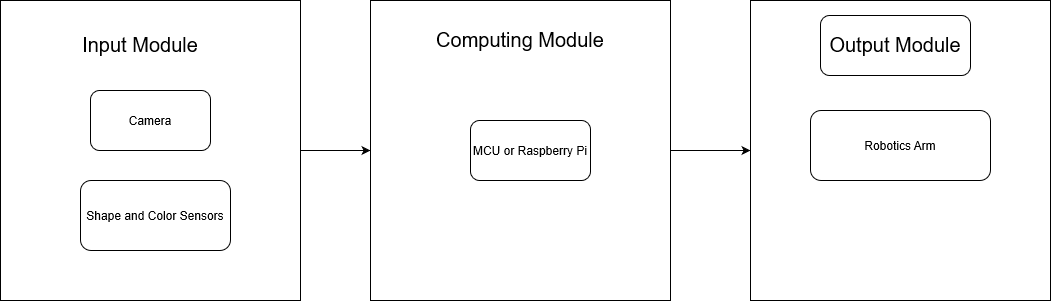


Figure 2, Robotics Flow Chart

Reference

1.Ftobler. (n.d.). Ftobler/robotArm. Retrieved September 19, 2020, from https://github.com/ftobler/robotArm

2.Juneja, R. (2016, January 5). Ball Tracking Robot. Retrieved September 19, 2020, from https://create.arduino.cc/projecthub/junejarohan/ball-tracking-robot-7a9865?ref=user