iSee

EECE 4320 – Digital Control Systems
David Helminiak

OPUS College of Engineering
MARQUETTE UNIVERSITY

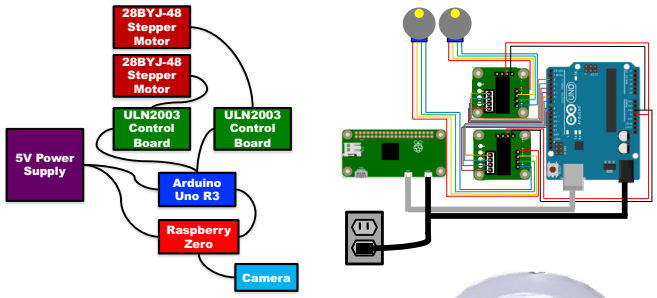
Description

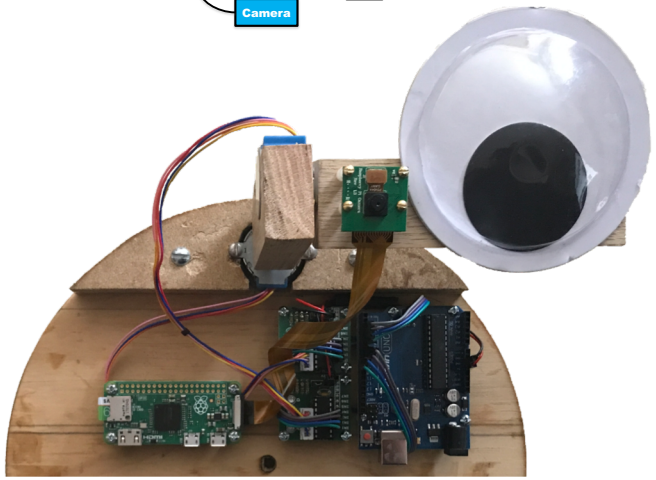
This project combines a basic understanding of a microcontroller's functionality with the openCV (Computer Vision) library to track areas of motion. A webcam, connected to a Raspberry Pi Zero, uses openCV to identify areas of motion and pass commands to an Arduino Uno. The Uno issues commands to a pair of motor control boards which then provide power to and rotate motors in both the x and y axes to point towards the general area of motion.

Parts

Component	Description	Price (US)
Pi Camera	• Sony IMX219 8-megapixel • 1080p 30 fps, 720p 60 fps, VGA90 modes • Uses the Picamera Python library Image: https://www.dexterindustries.com/shop/raspberry-pi-camera/	14
Raspberry Pi Zero	• Lowest cost single board pi-based computer • 1 GHZ single-core • 512 MB RAM Image: https://www.modmypi.com/raspberry-pi/raspberry-pi-zero-board-379/rpi-zero-board/raspberry-pi-zero	5
Arduino Uno R3	• Microcontroller board based on ATmega328P • 16 MHz clock • 32 KB RAM Image: https://www.sparkfun.com/products/11224	10
ULN2003 Motor Control Board	• Seven transistor drivers Image: http://www.mreeco.com/product/Uln2003-stepper-motor-driver-board-test-module-arduino/	4
28BYJ-48 Stepper Motor	• 5.625 degrees per turn • 34.3 mN*m holding torque • 5 VDC • 4 phase Image: https://www.tdegypt.com/product/stepper-motor-28byj-48-5v/	4
MicroSDXC Card 64 GB	• MicroSD card with 64 GB Image: https://www.amazon.com/gp/product/B06XX2959Q/ref=od_aui_detailpages00?ie=UTF8&psc=1	20
USB OTG to USB	• Adapter from USB on the go to USB 2.0 Image: https://www.amazon.com/gp/product/B00YOX4IU6/ref=od_aui_detailpages01?ie=UTF8&psc=1	3
Mini-HDMI to HDMI	• Adapter from mini-HDMI to full HDMI Image: https://www.amazon.com/HDMI-Adapter-VCE-2-PACK-Plated/dp/B0111HRJTO/ref=sr_1_5?s=electronics&ie=UTF8&qid=1522854327&sr=1-5&keywords=mini+hdmi+to+hdmi	3
Total		~$63

Design Concept & System Design



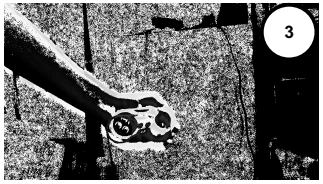


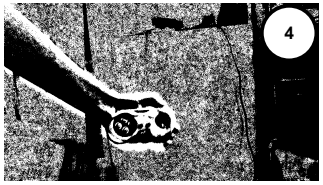
Motor Control Process

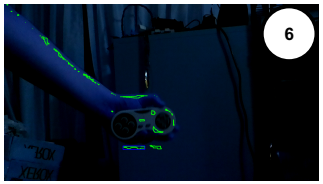
A base program uploaded through the Arduino IDE continually runs on the Uno waiting for commands sent through a the Pi's serial port. The Pi runs another script that compares a set of current and desired coordinates to determine what commands should be issued. If the Pi sends a clockwise command, the Uno then sends numbers 1-4 with the appropriate timing to the x-axis motor shield causing the motor to rotate clockwise. Reversing the order (4-1) results in counter-clockwise motion. Similar motion may be generated for the y-axis motor using pins 5-8.

Code

The code for this project was written with Python and is available on Github at https://github.com/Yatagarasu50469







Motion Tracking Process

The motion tracking process compares a pair of images to determine areas where significant levels of change have occurred. These areas are run through a rudimentary filtration process to identify the largest area of motion and produce a single coordinate of interest.

1-2. First a reset coordinate is set, such that upon completion the camera will move to its original location. Then a pair of sequential images are generated using a camera connected to the Raspberry Pi Zero. When initially being setup, the white balance on the camera is frozen to prevent light source changes from indicating motion. Since the images captured are mirrored, both are flipped.

3. The images are switched to grayscale to better isolate differences between them and then subtracted from one another.

4. The resultant image becomes converted into a binary format, where each pixel value only contains a 0 or 1 value. Areas in white now denote differences between the two images.

5. White areas that have small areas are further removed.

6. Remaining areas are most likely where significant motion changes have occurred and are labeled as individual objects. For visualization each area is surrounded by a green line, with a blue box placed on the largest area.

7. If the new coordinate is suitably different from the camera's current coordinate, then the appropriate commands are issued to the Arduino Uno. Motion is not tracked during the motor control phase, but resumes after its completion.