EE/SE/CPRE 491
MAY 1631
DESIGN DOCUMENT FINAL VERSION

Advisor

Leland Harker

Client

ISU Electrical & Computer Engineering

Project Title

Automated Tool Monitoring System

Group Member

Edward Droesch
Wan Zulsarhan Wan Shaari
Ibtisam Ostham
Mohamad Asyraf Samsudin
Deeksha Juneja

Table of Contents

Executive Summary

The goal of this project is to track down and secure the tools that are taken out of the toolbox. Often the tools go missing due to numerous reasons such as students forget to put the tools back in its original spot or take it out of the machine room for work and don't bring it back. Thus, we have designed a system that can keep track of the tools in the toolbox and eventually solve the issue of missing tools.

Project Statement

As the tools used in the ECPE machine shop often go missing or do not get returned to the toolbox, our group will try to solve the problem by creating a system called Automated Tool Monitoring System to secure and track down those tools borrowed by students. The system will allow access to the toolbox by the swipe of a student, staff or faculty member's ID card. Once opened, the system will keep video records of tools removed and returned. In addition to the video records, a vision detection system is used to keep a track of the number of tools in the toolchest after every transaction. The system will also interface with the lock system of the existing toolbox.

Ideally our system will utilize a single board computer called Raspberry Pi to handle the workload. As Raspberry Pi is capable of doing everything like a desktop computer does, we will connect a camera and user input components, which are magnetic card reader and keyboard, through USB port. User input components will allow the user to swipe their card or enter their ID card number. Servo will be used to unlock the tool drawer. Also, the system will use a 7' monitor as user interface. High resolution camera will be used to user activity when they log in to our system. Additionally, the system will be using National Instrument MyRIO which is programmed to do image analysis to detect which tool are currently missing from the tool box.

System Level Design

System Requirements

The system created for this project will meet the following requirements:

- The user with valid access should able to open the tool machine by using their ISU card or by entering their ID number using numpad.

- The camera for Main Control Unit should able to record 480p video for more than **3 minutes**
- The camera for Image Processing Unit should able to capture 720p image for image analysis.
- The Image Processing Unit should be able to detect which tool is taken or returned at more than **75% accuracy.**
- The Dropbox should have enough storage to store videos for 1 week assuming the number of tool transaction per day is less than 20.
  - 20 transaction * 7 days = 140 transaction
  - Each transaction takes at most 3 minutes of video = 420 minute worth of 480p video.
  - 1 minute 480p video requires approximately 50 MB storage
  - 420 minute * 50 MB = **21 GB of video storage**
- The validation process from MySQL database should take **less than 2 second.**
- The Raspberry Pi should able to be accessed remotely from any computer using SSH for configuration and troubleshooting.
- No more than one drawer should be opened at a time.
- The tool box must be physically locked.

Functional Decomposition

- Authorization
  - Read the identification number of the card swiped
  - Parse the identification number
  - Authorize the identification number with database

- Record Video
  - Activate camera
  - Record video
  - Upload video to Dropbox
  - Get a share link for the uploaded video

- Unlock Tool Box
  - Signal the servo
  - Unlock the key
  - Signal the servo again to lock the system

- Image Processing
  - Capture images
  - Feed the images to the Image Processing Unit (MyRIO)
  - Detect missing or returned tools
  - Send the data to Main Control Unit

- Detect Drawer
  - Get signal from sensors installed in each drawers
  - Send the data to database

- Web Application
  - Administrator
    - Access user profile
    - Monitor user activities
    - Grant and revoke user access to the tool box
    - See tool availability

  - Regular User
    - Request for access
    - See tool availability

System Analysis

- Main Control Unit (Raspberry Pi)

  Raspberry Pi is a credit card sized single board computer that can do anything that a normal computer can do. In this system, Raspberry Pi acts as a central processing unit that handles all the input from Magnetic Card Reader and Keyboard, run the main program and handle interaction with Dropbox, MyRio and MySQL database. It will also will interact other output component like monitor, speaker, camera and servo.

- User Input

  The two input component for this system are the magnetic card reader and keyboard. The magnetic card reader should be able to read the card number for validation.

- Camera

  There will be two camera that are going to be used in this system. The first camera is attached to the Main Control Unit to record activities of every transaction of taking or returning tools. The second camera is used to capture images for image processing and analysis to detect which tool is taken or returned.

- Lock Mechanism

  Servo is the main component for the tool box lock mechanism. Since there are two locks for each compartment of the tool box, there will be two servos used in the system. Each drawer of the tool box will be installed with a sensor to detect whether it is open or not. If more than one drawer is opened at the same time, the system will sound an alarm to warn the user.

- Image Processing Unit (MyRIO)

  The system will be able to detect which tool is taken or returned by the user. For the purpose of tool tracking, we have used the National Instruments MyRIO as the Image Processing Unit for the system. A few images will be captured by the camera and will be fed to the MyRIO embedded program and Vision Assistant script for image analysis. The result then will be sent back to the Main Control Unit for further processing.

  In Vision Assistant Script, we have used Pattern Matching functionality to detect which tools are missing from the drawer. We made a template for each tool and were able to match it against the tools taken or returned. The data about the tools is sent to the Main Control Unit so that the information about the transaction can be stored as an entry in the database and the website can be updated.

- Admin Web Application

  Admin will be able to see all the transaction records through this web application. It also enable admin to add new user to the system, remove user from the system, contact user and update database. Furthermore, both admin and users will be able to see the list of tools in the toolbox and their availability.

Block Diagrams of the Concept

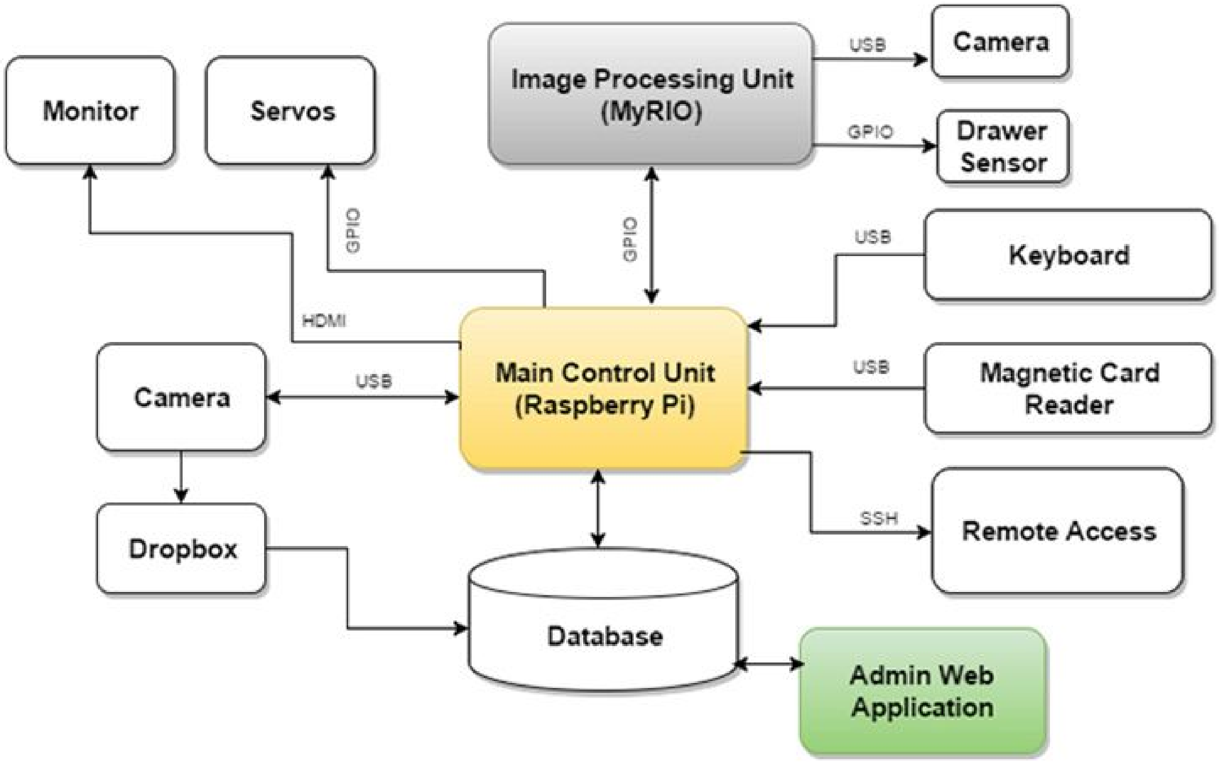


Figure 1 : Block Diagram

Detailed Implementation

I/O Specification

The input and generated output for our system is explained below:

Input:

- Magnetic Strip ISU Card Reader
    - Takes in the student/faculty ID number to log them into the Tool Checkout System
- Numpad
    - Alternative to enter the ID number to log into the system.
- Camera
    - Once the user has been authorized, the camera will be activated and record the user's activity.

Output:

- Sounding alarm
    - When user has more than one tool drawer open an alarm will sound to notify the user to only have one drawer open at a time
    - This is due to safety concerning due to a potential tipping over of the cabinet, should the weight be offset
- Servo Locks
    - After the user has logged into the system and are approved to remove tools from the cabinet, tool cabinet will unlock
- LED Monitor
    - Will display login screen as well as additional relevant information

User Interface Specifications

Our system interface shall be designed with human-usability in mind. The user will just simply swipe the id card to unlock the tool box. Onced swiped and approved access, our system will do the rest. Additionally, the Main Control Unit shall be constructed in such a way that the keyboard is easily accessible and typable. Besides, the screen will be set to have a large enough text to increase readability. The LCD interface itself shall be as simple as one input field for ID number, as well as relevant success/fail messages.

Hardware Specifications

- Raspberry Pi

  Raspberry Pi of model B, which comes with 512MB of RAM, 2 USB ports, an ethernet port, and composite video will be used for the project as the Main Control Unit. The processing speed and the RAM available has been tested and verified to be sufficient for the process to get done. Additionally, a 16GB micro-sd card will be used to store any additional relevant data(if any), and a usb splitter will also be integrated on one of the USB port to connect the keyboard, USB camera, and the magnetic card-reader.

- NI MyRIO

  We use National Instrument MyRIO model 1900 as the Image Processing Unit for our system. MyRIO has dual-core ARM® Cortex™-A9 real-time processing and Xilinx FPGA customizable I/O. It has a powerful processing power that is very essential to run a highly performance process like image processing and analysis.

- Servo Motor

  The motor being used to control the locking and unlocking of the lock is a Parallax Standard Servo (#900-00005). This servo has interface capability with PWM capable devices and can hold any position from 0 to 180 degrees. The servo can apply 38 oz-in torque at 6VDC and can operate in the range of 4-6V. The servo also has a maximum current draw of 140+/-50mA at 6VDC and draws 15mA under static conditions. The I/O of the servo motor is connected to the Raspberry Pi. The I/O of the servo is controlled by the Raspberry Pi. The servo is powered from 6V batteries, through a transistor, to conserve power. When the Raspberry Pi sends a signal to move the position of the servo, the increase in voltage causes the transistor to turn the servo on and move it to its desired angle. When the servo does not receive a signal from the Raspberry Pi, it stays at its current location and operates under static current conditions. The servo is independently powered from an external 6V battery source in an order to avoid current spikes.

- Magnetic Card Reader

  The magnetic card reader that will be used is Yasoo®New 3 Magnetic Card Reader. it is a new USB 3 Track Magnetic Card Reader fully plug&play compliant. It means that it will act like a keyboard. It is very light, and could be operated at a low temperature of 0 degree Celsius and as high as 40 degree Celsius. It is a USB powered device and didn’t draw high current and voltage.

- Magnetic Sensor

  MC 38 Wired Door Window Sensor Magnetic Switch will be used for the project. It is a small magnetic sensor with a distance accuracy of 15-25mm. The sensor is easily to be programmed. It consist of two wires which one is already defined to be connected to ground. The other wire will be connected to I/O port and the send signals to inform the Raspberry pi about the status of the drawer. Magnets will be placed on the drawer so that when one drawer is open, the magnet will be far and the sensor will inform Raspberry pi that the drawer is now open.

- Lock

  The Lock mechanism that will be used is the original lock that has already been installed on the drawer. We have decided to keep the lock mechanism the same as the original because when there is power outage or the system fails to work, the drawer could be resetted and operated manually. However, the lock mechanism will be integrated with servo so that when the system is operating, servo will act like a key to lock and unlock the drawers. The servo will be connected to a custom made rod. One end of the rod then will be connected to the original lock mechanism so that when the servo moves to a desired angle, it push the rod to lock and unlock positions.

- Camera

  Main Control Unit

  - For the Main Control Unit, we used Microsoft LifeCam Studio Webcam. It has a capability of 1080p HD widescreen sensor. And this camera also has advanced high-precision optics which support auto focus and technology that automatically delivers bright and clear video.

Image Processing Unit

- For the Image Processing Unit, we have used a Logitech Pro HD Webcam 1080P. It is a USB powered camera. It can take 1080p HD videos at 30 frames per second. It has a 78 degree wide view which is good for close range image capturing. For this project, we found that this camera would work perfectly with our system as it could easily take pictures with high resolution.

Software Specifications

- ### Raspberry Pi Software Specification

  The Raspberry Pi is loaded with Raspbian Operating System which has a similar environment with Linux OS. This allow us to write new code in the Pi text editor and compiled using Terminal. Raspberry Pi runs on the Broadcom BCM2835 system on a chip(SoC), which includes an ARM1176JZF-S 700 MHz processor and VideoCore IV GPU.

- ### Main Control Unit (MCU) Software Specification

  Most of the code for the MCU is written in Python. The main program is to handle the user input and interaction with MySQL database. Other than that, the main program will execute other Python program to interface with camera and servo. The programs imports several **libraries and modules** and runs several functions for the MCU to function correctly.

Libraries and Modules

- bcm2835 1.46
  - To access General Purpose Input Output (GPIO) on Raspberry Pi board

- avconv
  - To interface with the camera

- MySQL-python 1.2.5
  - To interface with the database

- Admin Web Application Specification

Client Side

- The client side of the web application is created using Laravel 5 framework which make used of HTML, CSS and PHP script along with Blade templating.

Server Side

- For the server side that is used to communicate with the MySQL database, we used PHP scripting language.

- Image Processing Unit (IPU) Software Specification

The image analysis program that is deployed into the MyRIO is created using National Instrument LabView and National Instrument Vision Assistant.

Vision Assistant is a software specialize in creating script for image processing and analysis program. It has a lot of built-in functions to help in analyze image. We essentially use this software to design our tools detection module. We use “Pattern Matching Template” from Vision Assistant and this require us to create a template for each of the tools type from the tool box. To assist the detection and image analysis, we use custom made foam with highly contrast color in the tool box. Below are some of the template that we use in the module:

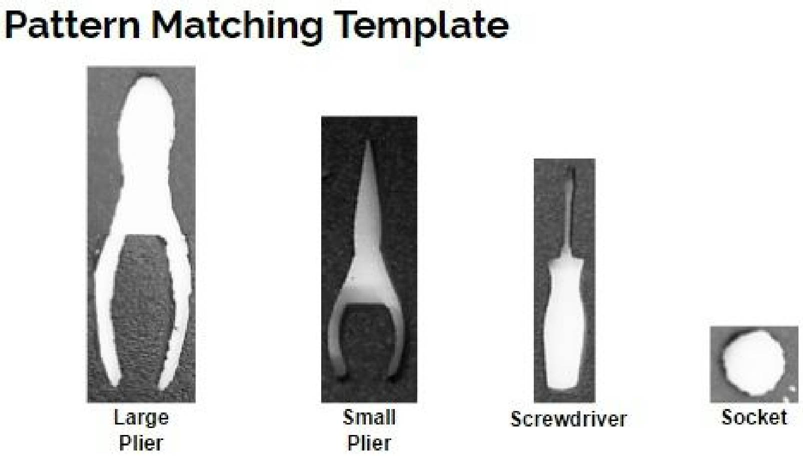


Figure 3 : Tool Template

Then we use LabVIEW to create a main program that will interface with the camera attached to the IPU. Besides, the main program also will also communicate with MCU to send information of how many tools are missing from the tool box.

- Communication Protocol between MCU and IPU

The communication between MCU and IPU is done using 5 Digital Pins. One of the pin is used by the MPU to signal the IPU whether to start or stop the operation. Meanwhile the other four are used by IPU to send the information back to the MPU. The only information that is sent back to the MPU is the number of tools missing detected from the transaction that has just been made by user. This number are send in binary through this four pin. Below is how the pin setup looks like:

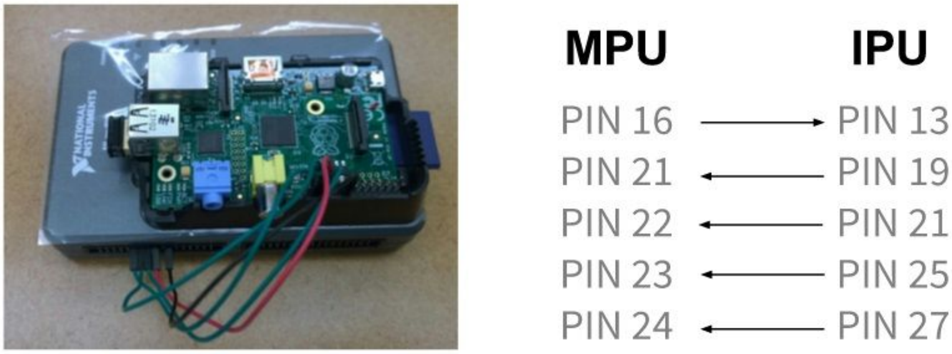


Figure 4 : Pin Setup

Implementation Challenges

- Communication between MCU and IPU

One of the biggest implementation challenge for us was to set up the communication between MCU and IPU. We tried various options like wifi - which was extremely unreliable to transmit the data and USB cable and ethernet cable - both were not supported by our system. However, in the end we settled on using digital pins. Digital pins seemed to be the best option because they are faster, and more reliable. However, they come with their own set of challenges. It is necessary to make sure that MYRIO and Raspberry Pi each is ready to receive the signal or else the signal gets lost. Furthermore, the pins require a tedious handshaking protocol between MYRIO and Raspberry Pi. Moreover, if one pin gets disconnected then the whole communication is cut off.

- Tool Detection Design

  Another huge implementation challenge for our system was to find the best way to detect the missing and replaced tools in the tool box. The initial plan was to use the MyRio for image processing and analyzing, as well as implementation of tool detection. However, the main problem with the current image capture system is motion blur. This motion blur renders the images taken, unusable for tool detection. There also exists the issue should the user accidentally, or intentionally block the camera's view of the tool drawers. This would lead to the system registering no tools taken from that user, when in truth, they have taken tools. However, this problem is out of the scope of our project and hence, remains unresolved.

- Lock Mechanism

  On the hardware side, a technical issue that we have is about integrating the servo with the lock mechanism for the tool box. The existed lock system for the tool box is quite mechanically complicated and none of our team members have much experience with mechanical lock system. However, we have figured out a way make our servo work with toolbox lock system. We have connected the servo to the lock system via a lever system as shown in the figure below. Hence, when the key is used to unlock the toolchest, the servo will rotate to unlock the drawer as well. However, now the user does not have to depend on the key. If the key is left unlocked, the user just has to swipe the ID to unlock the servo and the toolchest will be unlocked. This will provide user authentication to the toolchest and keep the tools from being used by unauthorized personnel.

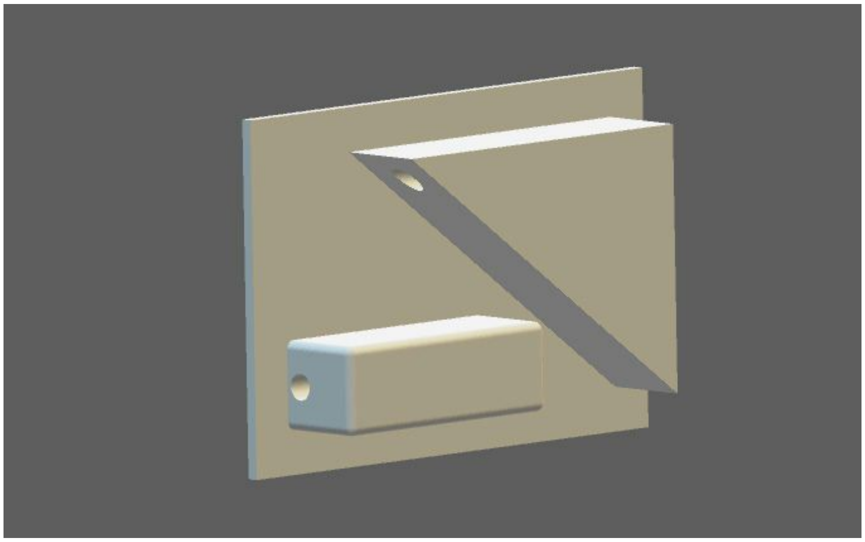


Figure 5 : Lever System for Lock Mechanism

Testing Procedures and Specifications

- ### Electrical Component Testing/Procedures/Specifications

  All electrical components such as servo and magnetic sensors are tested on breadboard first and then implemented into PCB layout. We tested the voltage supply for the servo to know how much power is needed in order for the system to work perfectly. Besides, the sensors are tested for its minimum and maximum distance of detection before the signal is sent to the Raspberry Pi.

  <u>Magnetic Sensors</u>

  - The main focus of the testing was to verify the integrity and functionality of the drawer detection sensors. These sensors must be able to generate a warning signal when more than one drawer is open. This warning signal will take the form of a signal being sent to the sensor's corresponding IO port on the Raspberry PI.
    - The testing procedure for this operation is simply confirming correct signals being reported to the Raspberry PI's IO Port. Along with the Raspberry PI's display and interpretation of the drawer status.

- Hardware Component Testing/Procedures/Specifications

  All hardware components are tested vigorously to ensure that they meet our high level system requirements. First, we tested the servo by sending a signal from the Raspberry Pi to rotate a corresponding number of degrees. Next, we verified that the servo moved the intended number of degrees in accordance to our needs. Then we did stress testing on the magnetic sensor by making sure that the Raspberry Pi will get the open and close signal from the the sensor. We also verified that the camera recorded a minimum length of 3 minute video and capture images at the same time.

- Software Component Testing/Procedures/Specifications

  All the software for this project are tested thoroughly including the Main Control Unit software, the Image Processing Unit software and web application. In particular, specific emphasis was given to test for security and secure information transfer from the system and database. This testing will ensure that no outside user will be able to get into our system, and gain unauthorized access. As for the image processing unit, we feed in different images into our Vision Assistant script and see if the script able to detect all the pattern in the images accurately. We also vary the distance of camera when we capturing the images.

  Image Processing Unit (IPU)

  - To test the accuracy regarding different foam colors, we used the “Pattern Matching Template” program from Vision Assistant software. The “Pattern Matching Template” relies on high contrast between the multiple foam layers to create a clear image to compare patterns against.
    - In our testing, we discovered that the combination of a yellow, base layer, with a black, top layer gave us the best results that remained within our accuracy interval.

  - Testing for accuracy regarding various heights is directly proportional to the quality of the taken image used for image comparison. To obtain the highest accuracy, we need to use a high quality camera for image capture. Once we acquired our PRO HD Webcam, we tested various capture heights with “Pattern Matching” Vision Assistant software. The final results showed an accuracy rating within the required specification.

Main Processing Unit (MPU)

- The main focus for the testing done on the MPU comes in the form of stress testing.
  - To do this, we generated random input from the IPU, and measured the reactions of the MPU by verifying its terminal window output.

Appendix

Appendix I : Operation Manual (for Administrator)

- ### Setting Up the **Main Control Unit**

  There are four main hardware component connected to the Main Control Unit and below are the details on how to set them up together

  - Webcam (USB)
    - Connect to any USB port

  - Card Reader (USB)
    - Connect to any USB port

  - Numpad (USB)
    - Connect to any USB port

  - Servo (GPIO)
    - There are two pin that is going to be used to interface with the servo which are:
      - Pin 12 (Pulse Width Modulation pin)
      - Pin 6 (Ground pin)

  - Image Processing Unit (GPIO)
    - The pin that are used for the communication with IPU are
      - PIN 16,21,22,23 and 24

  - Monitor (HDMI)
    - Connect to the HDMI port

- ### Setting Up the **Image Processing Unit**

  The image processing unit is basically only connected to two hardware component, which are webcam and the Main Control Unit. As stated before it is connected to the MCU through pins and setting up the webcam is just as simple as connecting it into one of the USB port on the IPU.

- Setting Up the **Tool Box**

  The existing tool box has to be set up with two additional components in order for our system to be fully functional and they are:

  - Foam for Tools
    - To assist the tools detection module
    - Just put the pre made foam inside the drawers

  - Lock Mechanism
    - The lock mechanism consist of 3 parts which are the servo, string and lever system
    - The set up is shown in diagram below:

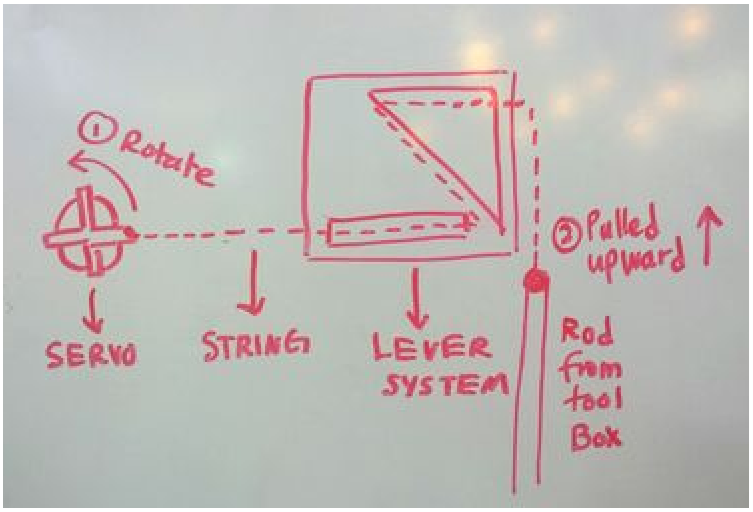


Figure 6 : Lock Mechanism Set Up

- Using the **Web Application**

  - See the list of users
    - Go to "**User**" tab
    - Click on any user name to see their detailed profile and their activity

  - See the transaction history
    - Go to "**Transaction**" tab
    - Click on any transaction to see more information about it and the video feed of the transaction

- Giving or revoke access from user
    - Go to user profile and change the permission type of the user by clicking the “**Approve**” or “**Revoke**”

Appendix II : Initial Version

- Tool Detection Design

The first design of tool detection module was to detect a sticker underneath the tools to decide which tools were missing from the drawer. The shape of the sticker that was used has circular and we were using circular pattern matching to detect that sticker.

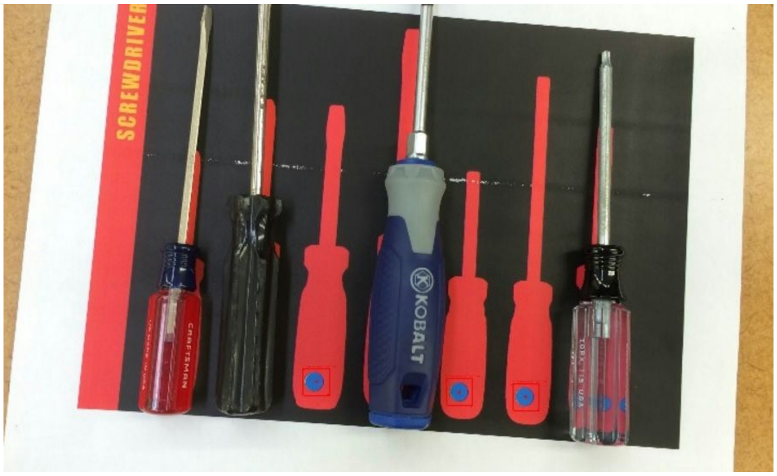


Figure 6 : Circular Pattern Matching Setup (Initial Version)

However, it would be difficult to know which which draw is open because of the marginal difference in the radius of the circular sticker. After that, we tried using different shape of stickers for each drawer. But then we found that this method will make our system harder for future implementation and expansion. Imagine that for a large tool box with more than 200 tools inside of them, the administrator has to paste a sticker for each of the tool manually for our system to work. After further discussion with our advisor, Mr Harker, we tried to make the Tools Detection Design more easier for future expansion. We looked into geometric pattern matching for each tool but discarded the idea due to low reliability and accuracy.

As for the current version of tool detection design, instead of detecting the sticker, we are trying to detect the pattern of each type of tools itself by using pattern matching. Even though we need to provide the Vision Assistant Script each of the tool's type a distinct template but it is better than to have manually put sticker for each tools. Moreover, it will accurately detect each tool that is missing instead of being confused about the shape of the sticker. We are converting the image to black and white and then implementing pattern matching to detect the tool missing.

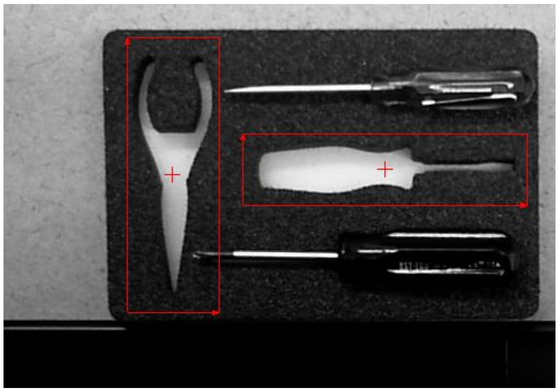


Figure 7: Pattern Matching Setup (Final Version)

Appendix III : Additional Information

- PCB Issues

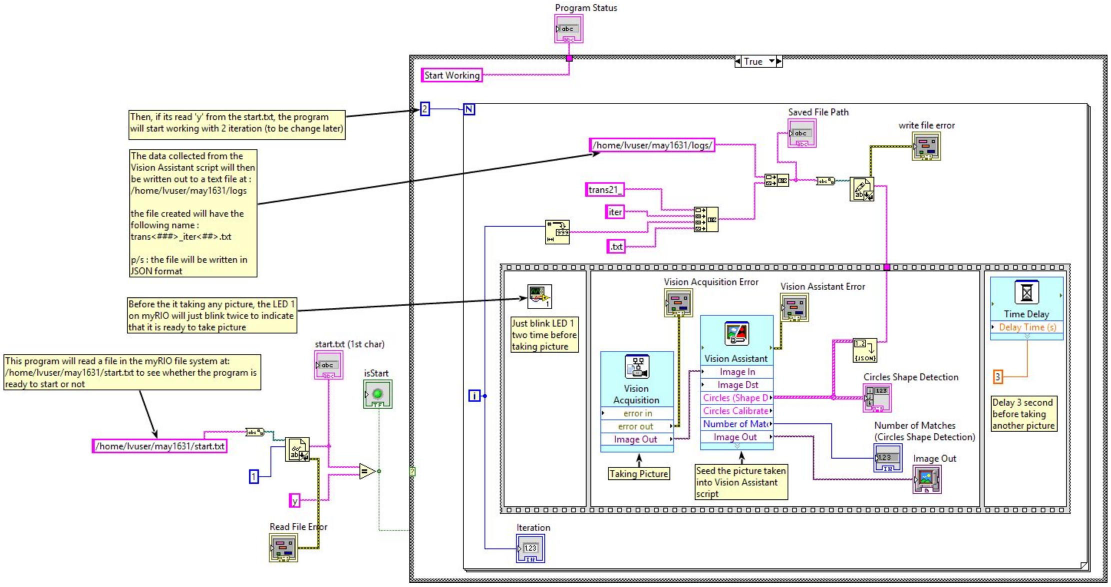
  When testing the electronic parts individually, we have encountered zero difficulties/issues except for the anomalous behavior of the parallax servo when left static at 0 degrees.

- Camera Exposure

  When we did image processing analysis in the Vision Assistant software by using one of the Microsoft webcam model, the images captured under went an over-exposure because of the room lighting and camera exposure feature. The pictures become too light making the accuracy of tool detection to drop. Thus, it is important to make sure the toolbox location is placed under modest amount of lighting and also making a good decision on camera selection.

Appendix IV : Gallery

- Image Processing Unit (MyRIO) Block Diagram



Then, if its read 'y' from the start.txt, the program will start working with 2 iteration (to be change later)

The data collected from the Vision Assistant script will then be written out to a text file at : /home/lvuser/may1631/logs

the file created will have the following name : trans<###>_iter<##>.txt

p/s : the file will be written in JSON format

Before the it taking any picture, the LED 1 on myRIO will just blink twice to indicate that it is ready to take picture

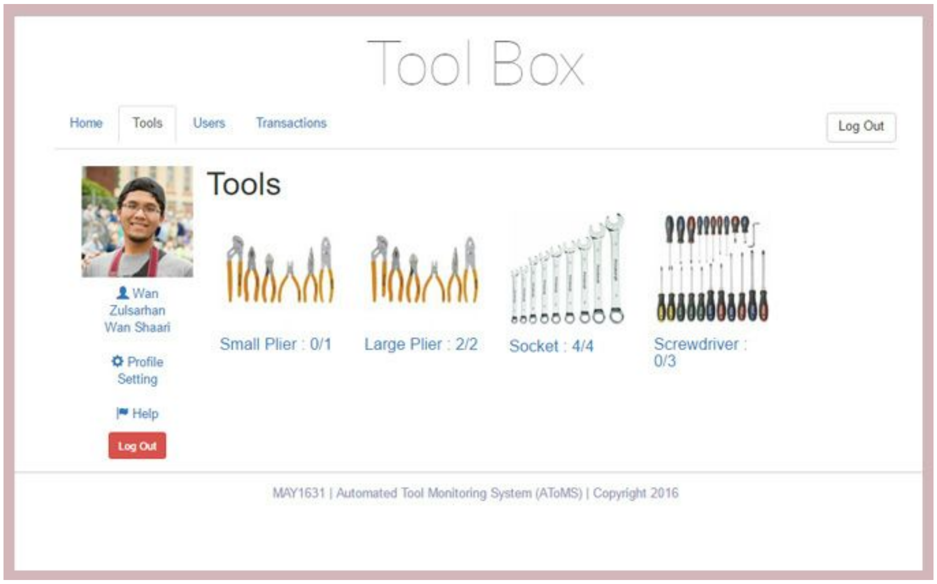
This program will read a file in the myRIO file system at: /home/lvuser/may1631/start.txt to see whether the program is ready to start or not

Just blink LED 1 two time before taking picture

Seed the picture taken into Vision Assistant script

Delay 3 second before taking another picture

- Web Application Screenshot



Tools

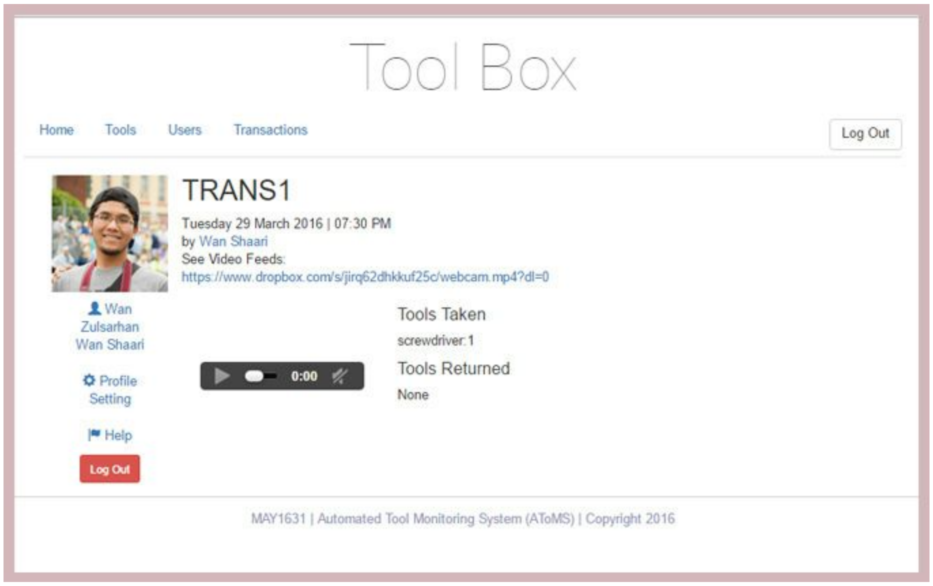
Small Plier : 0/1

Large Plier : 2/2

Socket : 4/4

Screwdriver : 0/3

MAY1631 | Automated Tool Monitoring System (AToMS) | Copyright 2016



TRANS1

Tuesday 29 March 2016 | 07:30 PM
by Wan Shaari
See Video Feeds:
https://www.dropbox.com/s/jirq62dhkkuf25c/webcam.mp4?dl=0

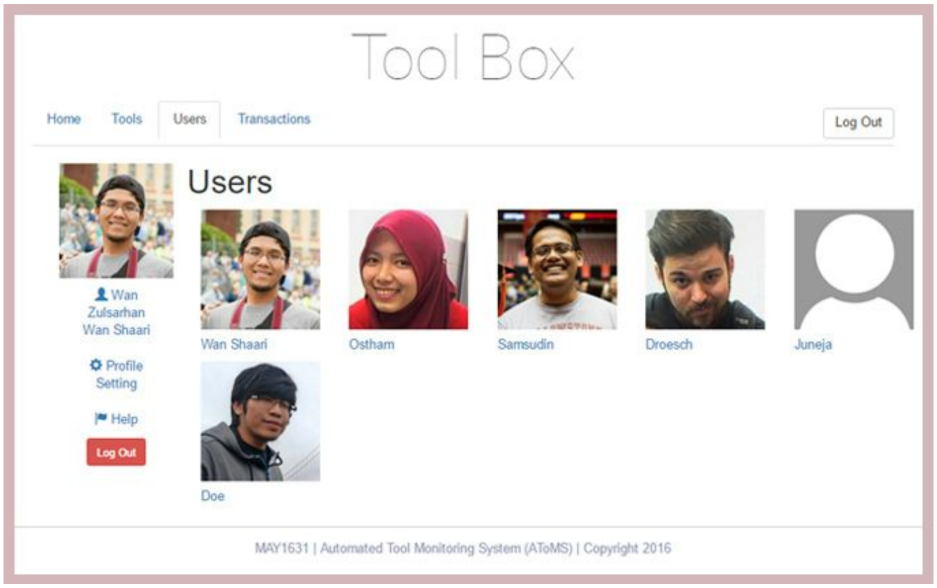
Tools Taken

screwdriver:1

Tools Returned

None

MAY1631 | Automated Tool Monitoring System (AToMS) | Copyright 2016



Conclusion

Our main goal was to create a system that would benefit the ECPE Department. We are creating this system with the hope that anyone who take tools from the department`s tools drawers will feel responsible to return the tools back to where it belong so that it could be used again by others. As our work was focusing on creating a security for the tools, we have created a system to monitor who took which tool and when. Also, we will record a video of them taking the tools out from the drawer. By using the camera, we are capturing images of the drawers every second to match them against the template to check which tool is taken or returned. Then the information about the transaction is stored in the database. We have a prototype ready which can be easily extended to the whole toolchest. Our product is a cheaper alternative to the products already in the market. Which makes it affordable for implementation at the university level. This project is a success not only in terms of the product and cost but also in terms of expanding the knowledge base of the implementers.