12-6-2022

Industry 4.0

Zachary Zeitler
Western Michigan University

Senior Design Project:
Industry 4.0- Digital Twin

Josh Getsinger – Jaden Perrine – William Sinn – Zachary Zeitler

Table of Contents

Problem Statement

Need

The process of manufacturing (printing, machining, welding, etc.) requires some level of quality control. Without any vision system, a machine operator must take the time to measure the part with separate equipment. There are also several ways operators and machines can make a mistake during the manufacturing process, resulting in product defects that would be managed by quality control. The need is to improve quality control efficiency by creating a vision system that can perform the measurements without manpower, extra machines, or time.

Objective(s)

The objective of our client is to automate the quality control process, thereby significantly increasing machine efficiency. Our objective was to build the first piece of software that can be used to interface with cameras and a laser. It uses cameras and laser to generate images and 3D models of the part being produced, which can be used to take measurements and define changes for the machine.

With our client, we decided not to include the actual taking of measurements as part of our requirements. Our limiting of scope removes layers of complexity and provides us with the appropriate amount of time to have successful results that will be realized by future groups using our software.

Terms, Acronyms, Glossary

PCM: Point-Cloud Model
UI: User Interface
TKinter: Standard Python Interface Toolkit (TK Interface)
OpenCV: Python Image Capture Library
Bot: A program that performs a series of steps to run another program
Arduino: Brand name for our microcontroller hardware

Problem Analysis and Research

Interface

Dr. Pavel showed us a prototype scanner model, which was meant to run with an open-source program. This was a 3D scanning software that used a rotating table and fixed laser to scan an object. With our client, we explored many open-source scanning programs, mostly which were not complete, or were in a foreign language. These programs had an interface window, so it was natural to follow this format. We proved successful in implementing a python interface that provides the client with the functionality the requested.

Image Capturing

The first thing we needed to know was what is to be captured. The client requested to have each exposed face captured. This means a simultaneous image capture from the front, back, left, right, and top face of the part – five cameras. More USB ports would be needed to manage these devices appropriately for the user. We purchased a powered USB hub and wrote a piece of python code incorporated in the interface to find and label the cameras. The interface gives the user the ability to configure the cameras, assuring they are in the right order.

3D Scanning

After analyzing several methods of scanning, we set the goal for one where the worktable and the camera/laser could be fixed. This left us with a sweeping laser on a stepper motor, and a running open-source software called MakerScanner. We were unable to compile the program and extract the scan function from it to stitch it into our interface. Instead, we set up a bot that will open MakerScanner and get through all the pop-up boxes to process a scan, then close.

Machine Signaling

One of the important pieces of our software is the ability to build off it, towards the goal of having it running in tandem with machines. Future groups will be trying to use our functions as part of the machining process. This means that the scan and capture function will be called by the machine in its own code, and it will have to communicate with our interface. The Arduino is listening for signals and can call the scan and capture functions when it receives the signal. Our interface will return a “scan complete” signal so the machine can move to its next operation.

Requirements

Functional Requirements

Image Capture

- Create a set of images from the connected cameras
- Save image sets in user-specified destination with appropriate naming conventions

3D Scan

- Linear Scan
- Create PCM
- Save to user-specified destinations with appropriate naming conventions

Automation

- Scans and image captures should be able to be triggered by machine signal

Scan Quality

- Either scans need to be cleaned by a noise/artifact reduction algorithm or noiseless scans need to be taken

- Scans should be of as high of a resolution as possible, .001” is preferable

User Interface

- Let user view scans/images
- Allow user to configure camera positions (top, right, left, etc.)
- Give user the option to select photo directory

Nonfunctional Requirements

Hardware Components

- Cameras
    - Must have five with mounts and a USB hub
    - 720p cameras have been provided by client
    - Four cameras are fixed at the side faces
    - Fifth camera is to be mounted to toolhead
        - Used for top face image and scan process
- Laser(s)
    - One blue laser, attached to a stepper motor with a custom wood piece
    - Must be positioned in reference to the scanning camera (7.8” apart)
    - Motor is controlled with Arduino scan process
- Frame
    - Wooden
    - Wrapped around the specified machine
    - Four ¼" camera mounts on each face
- Wiring
- Power Supply
    - Custom power control board for laser control
- Relays/Control Components
    - Using relays, have our software read start signals and return end signals
- Arduino Uno/Associated electronics

Software Components:

- Design functionality
    - View images
    - Settings for cameras
    - Settings for laser
    - Manual scan/capture buttons
- Design Interface
    - Check boxes to choose which cameras will be taking the images
    - Button to open file dialog choosing where to store images
    - Images to show the last image captured by each camera (or a placeholder)
    - Button to take images on all selected cameras

    - Main page
    - Directory selection file dialog
    - Camera configuration window
    - Scan configuration window
- Implement functionality using Python
- Tkinter window using Python for back end
- Use Arduino to communicate with machines for details about controls
- README

Standards and Constraints

Applicable Standards

Our program has the minimum requirement of being fully compatible and functional with windows machines. There is a secondary optional requirement of having the software be cross-platform compatible with Linux systems as well. This is something the client would like but not one that is required.

Our client has a plethora of machines he wants to implement this technology on. Our design should be one that can be worked to fit a variety of machines. To get things started, we decided to test on a router in the lab. The tooling that holds the cameras is out of scope, we only need to build what is necessary to test the functionality of our software.

There were no specific requirements from the client for the implementation. Several open-source scanners were suggested by the client to study, which led us to our design. However, we self-imposed the general requirement of making as much as of the program as cross-compatible and modular as possible.

Future modifications are expected to be made to the interface and software, including but not limited to the addition, removal, or modification of functionality. We will be leaving our software in the hands of the engineering department to with as they please.

Constraints

Costs

We had only our personal budget for this project, and there were a few expenses. We needed extra USB ports, something to hold the cameras in place, power supplies, stepper motors, and functional scanning software.

Scope

As the computer scientists working on a manufacturing project, we wanted to leave work the other department would be proficient in and take on the work that would help them where they might be lacking. Focusing on creating the software that works with the cameras leaves the opportunity for future teams to design the mechanical parts that our software will be compatible with.

Our project doesn't need to run in sync with operations. We also left measurement taking out of our scope.

Time

We spent the first semester understanding the meaning of our project and establishing scope. Our second semester was spent creating and implementing. We met weekly with our client, who was excited to share with us anything relevant to the project and was happy to help us in any way. We had an additional team-only weekly meeting in the lab, which our client would typically stop by for a word.

Ethical Analysis

Identify moral issues

The moral issue is to determine if our software is safe and beneficial to the WMU department of manufacturing. This project is simple in nature, and the functionality and safety can be measured.

Identify additional facts helpful to making the decision

Our software runs on a dedicated machine. The machine does not run while our cameras are stationed in place. Our software takes image sets and point cloud models of an object being produced by a machine. The cameras will only be capturing images of the object and potentially some background. The cameras face inward towards each other. No audio is captured. We have relatively no budget.

Identify available alternatives

Right now, parts are manufactured, then taken to a measurement area manually. This software aims to eliminate that slow process. The only reasonable way to make a better version is with a budget and more time.

Identify personal impacts to the decision maker

Dr. Pavel will decide whether to use this software or not. He is designing a smart-factory class, as well as other projects that will rely on this technology. He is prepared to have other students from different departments take our software and build on it. Our software needs to be able to perform the functions he has specified and in a free and compatible way. If our software doesn't work, is too slow, or if it poses a risk to his equipment, he may decide not to use it.

Apply ethics theory to the problem

Hosmer Theory

We set out to create software that accomplishes a specific task without adding any additional problems. Problems may include cost, time, safety, or privacy. We have created a system that did not require much capital at all. Our system takes time to process but will be significantly faster than the existing process. Our system does not create any safety or privacy concerns.

There are no legal concerns. We used open-source software and will leave our project open-source without any financial expectation.

It is the duty of Dr. Pavel to continue to develop his department. This project we have for him is a step in the direction of development, thereby fulfilling his duty.

Analyzing our project through the lenses of Hosmer leaves us confident this is an ethically sound project.

Shareholder theory

The shareholders are Dr. Pavel and his students. Dr. Pavel has long-term objectives, and he will have other students working with or on our software. Our software will give these shareholders the ability to take controlled image sets and scans of a controlled work environment. Our software will not give the user any access or abilities they aren't already authorized for. The code is readable and with appropriate comments, so future developers don't struggle with it. Given the software works, it would be ethical to use it.

Stakeholder theory

The stakeholders include Dr. Pavel and his students. It also includes our team and Western Michigan University as a whole. Our software will be useful to Dr. Pavel and his students directly as they will be able to use this software as a tool in their own projects. Dr. Pavel is developing several courses that will use our software. This reflects well on the department and on the University as a whole. All the stakeholders will benefit from this.

Utilitarianism

The software is a tool meant to assist the inspection process. There are no consequences or risks from using this software. Therefore, any good brought by this tool is not countered by any negative collateral results. The utilitarianism perspective supports this.

Rights

No rights are at stake. Dr. Pavel has the right to use and experiment with technology. Our software uses a piece of open-source code, but open-source software is available for any to use however they want. The right not to use our software will also remain.

Justice

Those in the industry of creating and designing are always looking for tools to make it easier. Automation of inspection is a concept that almost only benefits manufacturing. The primary consequence is potential redistribution of labor. This is not a legitimate concern for the manufacturing department of WMU, and our project alone will not broach that consequence.

Virtue Theory

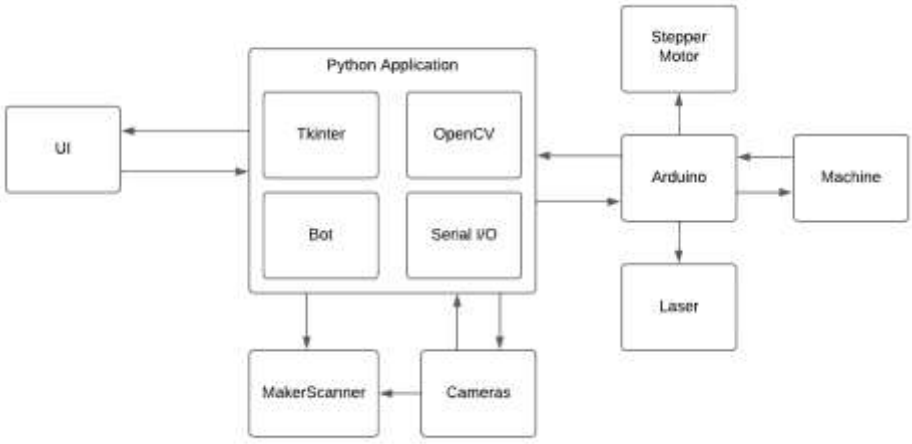
It is better to work more efficiently; it yields better results. Developing tools to help us work efficiently is a noble cause. It is therefore virtuous to use this software.

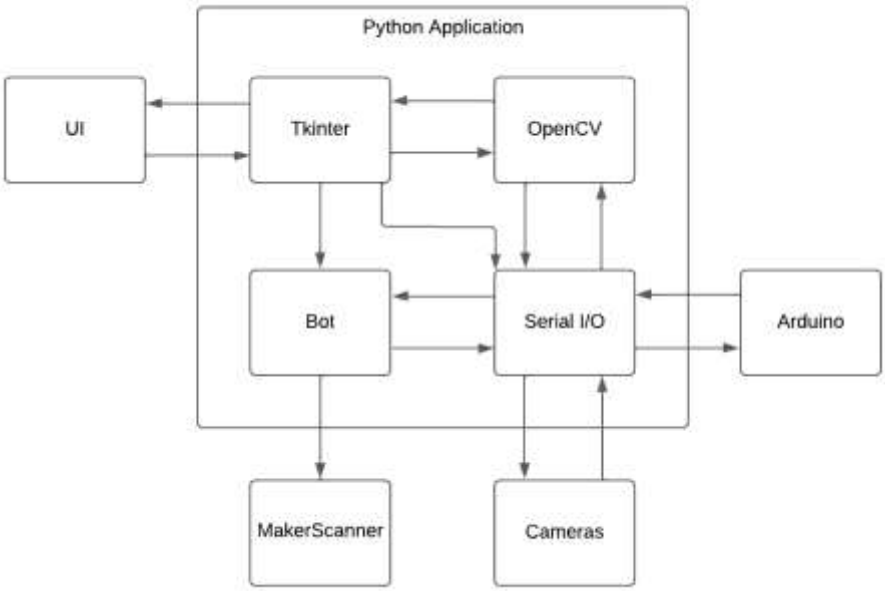
Conclusion

In conclusion, there is not any apparent moral consequence from using this software. It is perfectly safe to use this as intended. This conclusion was unanimous from the stakeholder theory, shareholder theory, and virtue theory perspectives.

System Design

Before discussing the structure and design of our system, we feel it is prudent to discuss the structure of its diagram. You will see that our diagram consists of various rectangles, or components, all connected between various arrows, or connections. The components themselves are aspects of the system, both concrete aspects like the Arduino or Tkinter and more general/conceptual ones like the UI. The arrows that connect them, on the other hand, reflect lines of communication between from one component to another. These communications consist of both instructions and simple data. We will be discussing the connections between each module, generally and in depth, so please use the diagrams as reference.





UI/GUI

The UI is entirely created and managed by Tkinter. In a very concrete sense, Tkinter is the UI. However, TKinter does much more than UI management; we will discuss more in the section dedicated to Tkinter. Because of this we found it is prudent to segregate them when describing the system's design for the sake of understandability and better separate aspects of functionality.

The UI allows the user to manage all the functions of the system. These functions consist of a main window screen, camera configuration screen, scan configuration screen, and change file directory button.

Main Page

The main page acts as the host for the methods the user will use to access the program's functionality. It consists of buttons to edit camera or scan configuration, a location to display images captured by the program after they are saved, a set of checkboxes that the user can select to choose which camera positions they would like to capture, and buttons to initiate the scanning or image capture process.

Camera Configuration Page

The cameras can be plugged in any order on the USB hub, but our program cannot make any assumptions on the order they are registered regarding the 0-4 index. It is necessary to have some method for the user to specify a position configuration for them. The Camera configuration page does exactly this, allowing the users to map each camera to the appropriate position. This page consists of a series of locations to display images captured by the cameras along with their corresponding position labels.

Upon opening this configuration window, images are captured automatically and presented in a basic default order based on the order the devices were recognized. Beneath each location is a dropdown menu that the user can utilize to map the camera to its correct position. The user can manually make a new image capture if needed. The configured values are stored in a dictionary which is saved in a pickled file in the same directory as the program.

Scan Configuration Page

Like image capture, the scan process requires certain parameters that a user must specify. This requires another configuration page. Our scan configuration screen consists of a series of sliding bars that can be used to specify values on a scale from minimum to maximum value. This is to prevent invalid values from being passed by the user while also minimizing the need for repeated validation checks. The variables are angle, speed, camera number, number of sweeps, and distance from object.

The only input value that is not a sliding bar is distance from object which must be specified by the user (in centimeters) in a text field. As this is text input, it is necessary to validate it, which is done by a simple typecast to integer and an exception clause to display an invalid input error when typecast to int is not successful. In the same manner as the camera configuration values, these values are stored in a dictionary which is saved in a pickled file in the same directory as the program.

Directory Selection Window

This window is far simpler than the others as it simply consists of the OS's specific directory selection window. Once the user selects a directory to save images and scans to the location is saved in the program. If a null directory is passed, the program throws and displays an error in the main page and doesn't save the location.

Tkinter

While the UI is essentially the dominant aspect of the Tkinter, its other functionalities are also critical to functionality. This is because the buttons in the UI call their associated functions on separated threads of execution. These threads are entirely spawned, managed, and closed by Tkinter. Tkinter also allows for other aspects of the program to modify the UI, such is the case for the camera capture that both saves images and displays those same images to the user in the UI. The Tkinter main window can also be used to store variables that can be later retrieved from a different scope when it is passed as a variable. This allows for easy modification of the main window despite differing scopes.

OpenCV

OpenCV is an image library for python. It is used by the program to both capture images and to detect connected cameras. Because OpenCV does not actually have a dedicated method to find all connected cameras, the simple solution we use is to iteratively check all camera ports registered by the OS until the program fails to find one. This "brute force" camera search is simple but has the potential to fail easily. There wasn't another solution we could find within

the time constraints of our project. The image capture process can be triggered both by the user through a button or by serial input from the Arduino. Once triggered the process starts by iterating through all selected cameras and capturing images from all of them. These images are then saved to memory with their mapped position and current time as their titles. They are also displayed in the UI so the user can preview the images that were captured.

Bot

The bot takes the information from the scan configuration specified by the user and uses it to navigate and input the set values into the MakerScanner UI. It then starts and stops MakerScanner at the appropriate times shortly before closing the program. The bot execution is invoked by Tkinter but it does not at any point communicate back to Tkinter once invoked. The bot can be triggered through Tkinter both by a button press or by Arduino signal.

Serial I/O

Serial communication in the program can be split into two distinct categories, camera I/O and Arduino I/O. In the program both are separate and distinct. Camera I/O is handled entirely by OpenCV. Everything from image communication to image processing and saving is abstracted by OpenCV so at no point are we directly handing camera control. Similarly, the camera capture process occurs on a sperate thread of execution but that is handled by Tkinter so we are not directly managing any camera threads.

The Arduino I/O however is entirely explicit. We are sending and receiving information directly to and from the Arduino in the form of utf-8 strings that encode specific instructions to and from the program. Because the program needs to always be listening to the Arduino for instructions, we needed to have a separate thread that we could explicitly call and control whose sole purpose is to listen to the Arduino. This thread was implemented as a daemon of the parent process so that there are no problems if the main program crashes or is terminated. The thread is spawned and given the main window instance as well as other critical information so it can use window functions to modify the main window while still executing separately and concurrently.

Some who are familiar with the pitfalls of multiprogramming might fear the possibility of corruption of shared resources. This, however, is not likely to occur as the Arduino thread itself executes the image capture and scan processes when their signal is received. This means that it is no longer listening to the Arduino and any instructions from it that might cause issues with shared resources, say for example two immediately subsequent image capture instructions, would not be possible because the instructions would simply be ignored. It is still possible in theory for these issues to occur when running an image capture from both the UI and the Arduino simultaneously, however such a case is so unlikely that we assume that it will not occur and leave it as a task for a potential future refinement.

Cameras

The cameras in the program serve two purposes. One is to facilitate image capture and the other is to facilitate the scanning functionality. Both features require some level of configuration. For image capture, we need a mapping of cameras to positions. This is done entirely on the user level. On the system level, the cameras remain the same, instead, their "camera index" numbers are stored and mapped to a position in a dictionary object. For the scan, all that is needed is the index of the camera that will be used for the scan process. The user can select this on the scan configuration screen.

MakerScanner

MakerScanner is an open-source software used to generate point cloud models using edge detection algorithms on input from a laser scanning process. In essence, the program takes a feed of video and takes a set of parameters to determine the normal displacement of the laser to the background. Then, using edge detection, determines the displacement of the laser from by the object being scanned. This displacement is then used, roughly, place three dimensional points at the positions detected. The result of this is ideally an accurate scan of the real-world object. MakerScanner requires parameters to be set by the user each execution. These are configured by the configuration page and delivered by the bot program.

Arduino

In our system, the Arduino performs two primary functions: enabling communication between our program and the machine and controlling both the laser and the stepper motor during the scanning process. When the machine is ready to begin capturing images or scanning the object being manufactured, it can manipulate the voltage of the corresponding pin on the Arduino using a relay. The Arduino then writes a word related to the desired process over the serial port to our program; "img" is sent if the machine triggered an image capture, and "scan" is sent if the machine triggered a scan.

Due to the nature of the serial communication between the Arduino and our program, all words sent from the Arduino to our program must be decoded using the UTF-8 method, and all words sent from our program to the Arduino must be encoded using the same method. In the case that the word "img" is written to our program from the Arduino, our program will perform the image capture and then send the word "FinishCapture" to the Arduino. The Arduino will then send a signal on one of its pins to a relay which informs the machine that it may proceed with its process.

In the case that the word "scan" is written to our program by the Arduino, the bot begins the scanning process, writes the appropriate values from the scan configuration dictionary to the Arduino, and writes the word "Start" to the Arduino. On receiving the word "Start", the Arduino turns the laser on and activates the stepper motor to begin sweeping the laser back and forth across the object being manufactured. Once this process has been completed, the Arduino writes the word "done" to our program, which causes our program to finish the scanning

process, and sends a signal on one of its pins to a relay which informs the machine that it may proceed with its process.

Stepper Motor and Laser

Both the stepper motor and the laser are controlled by the Arduino, with a one-way line of communication going from the Arduino to each of them. The Arduino controls the stepper motor according to the values of “Angle”, “Times”, and “Speed” from the scan configuration dictionary, and it does this only during the scanning process. “Angle” sets the total angle through which the laser will rotate on each of its sweeps across the object, “Times” sets the number of times that the laser will sweep across the object, and “Speed” sets the rate at which the laser will sweep across the object. The laser is simply activated by the Arduino at the start of the scanning process and deactivated by the Arduino at the end of the scanning process.

Machine

The machine interacts with the system only when it is ready for images to be captured or for a scan to take place. All the machine needs to do is send a signal on the pin corresponding to the correct action, whether image capture or scan, and then wait until it receives a signal on the correct pin indicating that the image capture or scan process has completed. After receiving such a signal, the machine can continue its operation until it is again ready for image capture or scanning to occur.

Testing

Testing of our software is done manually through the usage of the GUI we created for the application. The reasoning behind this is twofold: firstly, the process for testing is simple and tangible when done manually; secondly, the process of creating an automated testing suite would be both difficult and possibly inaccurate in its results.

For example, let’s look at our approach to testing the image capture. Whether or not our code works can be seen by having the images come through to our software. However, performing the same process programmatically is rife with issues. It would be costly in time to develop an automated test to verify something we can see with our own eyes instantly. This was also the case for other parts of our program when we discussed testing.

Even seemingly simple tests, like checking if a capture saved to disk is accurate to the capture made in the program are deceptive. We might think to save the image to disk using the capture function and then to test it by loading the saved image into the test and compare it to the one created by the function to see if they are the same. If we were to do this and the image were corrupted, the test would still pass as both the function's image instance and the instance loaded from disk would be corrupted. This deceptive result could cause us to conclude falsely that the program is performing correctly even though it has encountered an error.

Our ad-hoc and informal testing method is lacking regarding thoroughness, when compared to a built-in test suite. However, when considering the difficulty in implementing such a suite, as

well as the tightness of our time constraints, we concluded that it would be best for our group to focus on functionality and simply test manually.

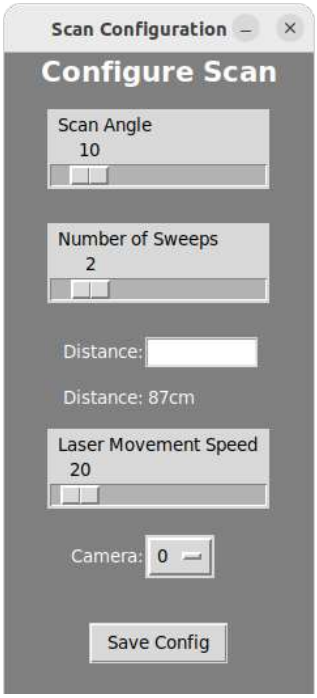
Results

Realization of Requirements

Our team was able to create and demonstrate our new software for our client. Our project accomplishes what was agreed upon with our client and sets up future groups to take it further. We can perform the image capture and scan process with a button from the interface as well as through machine signaling, which is how the process would be automated in the future.

Main Interface Page

Camera Configuration Page



Scan Configuration Screen

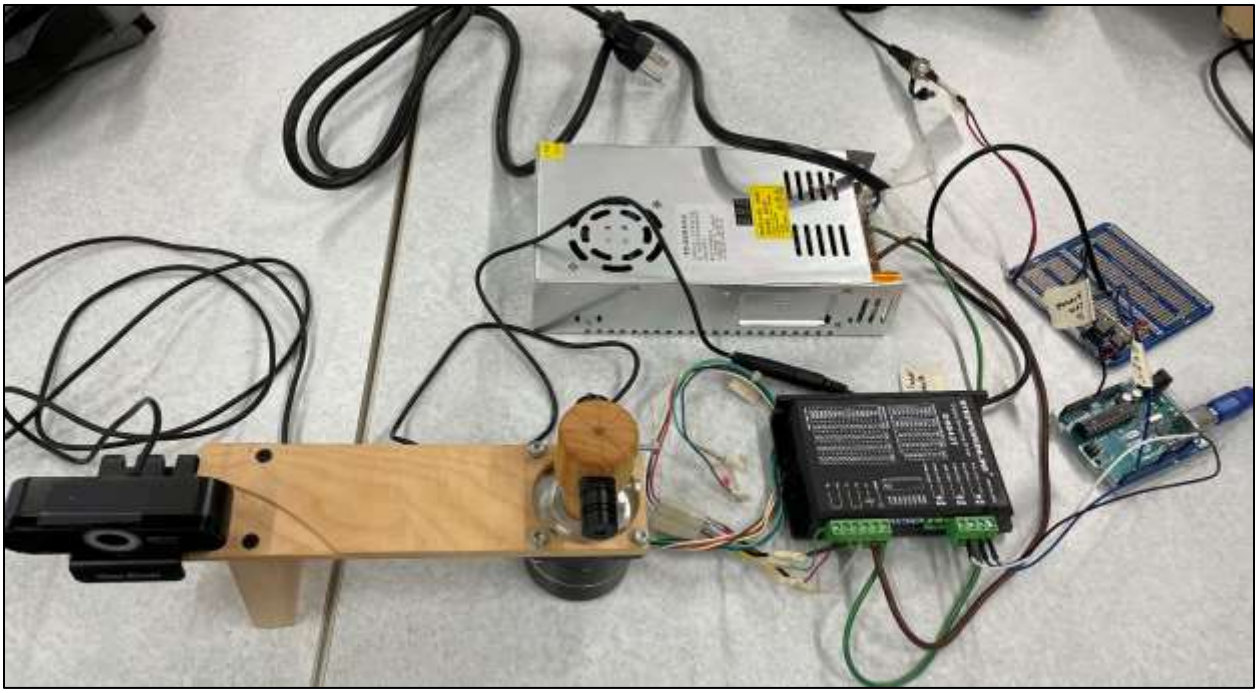
Realization of Standards and Constraints

Our program managed to achieve the minimum standard of being able to be fully functional in a windows environment. However, the optional, secondary requirement of being cross compatible with Linux environments was not achieved. For Linux machines, everything up to the scan functionality can still be achieved, but as the scan functionality itself requires software that does not easily compile on Linux machines.

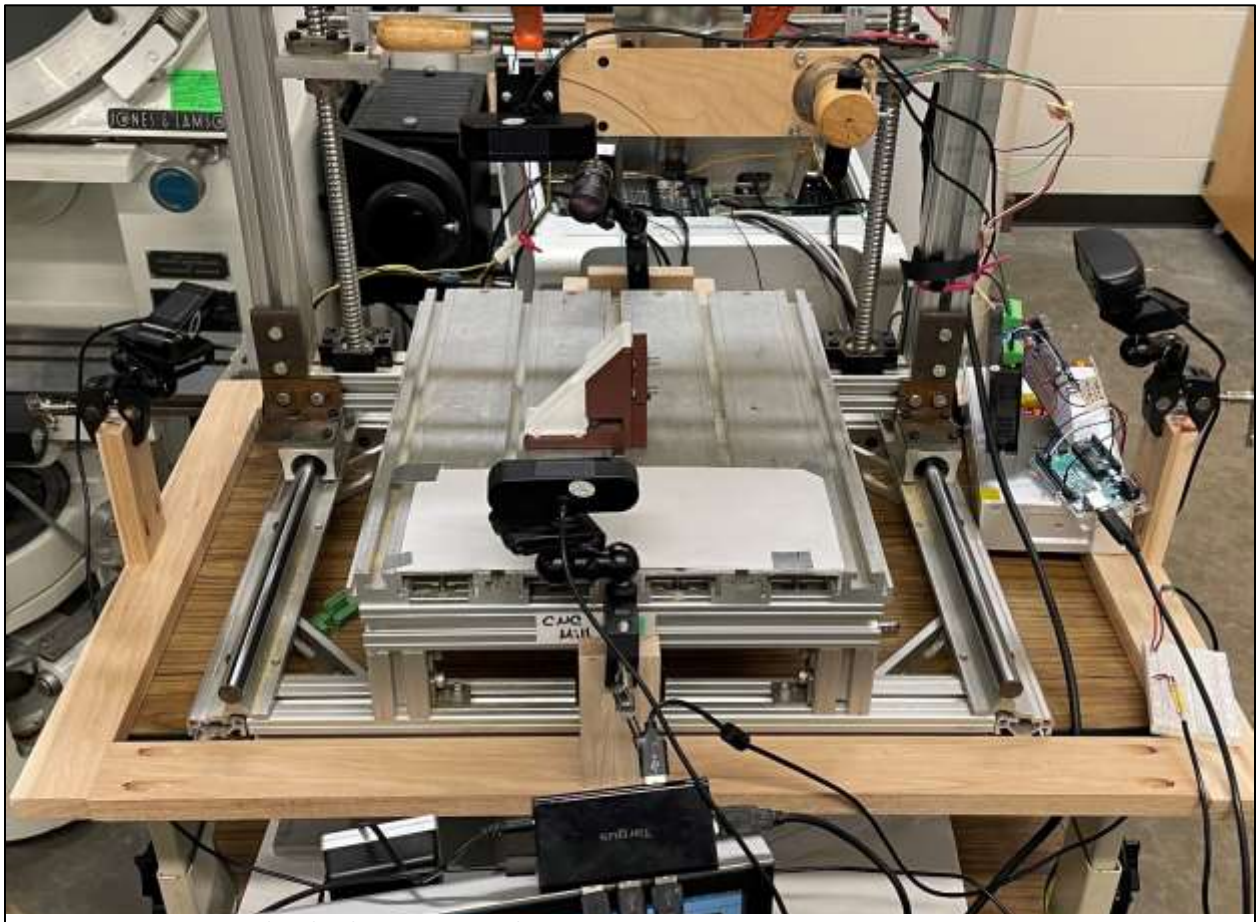
Out of all the scanning software we studied, we went with MakerScanner. We were able to compile it and control it with the bot to utilize the scan function.

We went out of pocket for the USB ports, camera mounts, and parts for the laser power supply. The cameras and laser were provided by our client. We also had to wire up a power supply system for the motor and the laser using personal tools and parts.

The wooden frame and laser mount were made with hardwood in personal shops. These were designed around a router in our client's lab. Images are attached below of all the hardware we used.



The scanner hardware: A camera and laser mounted to wooden bracket. The laser is on a stepper motor that pivots for the scan. The Stepper Motor Driver is used between the Arduino and motor. The laser and motor have power supplies.



Our vision system mounted to the clients router machine.

Testing Results

Our software is simple in nature, and the functions are easily tested. The environment in which it's being used is very controlled. Our scope control limited the number of variables we needed to account for, so our software can be tested simply by using it. If it doesn't work, we won't get the results of saved images and point-cloud files. If we are getting the images and point-cloud files, then our software works. We got the images to come through and have control over the devices in relation to the functions.

Future Work

There are two directions to go from here. Firstly, automating these functions into a machining process will require more agile and industrial tooling. The new tooling would be a frame holding cameras the toolhead can pick up, put into position, then set back down to continue machining. When the tooling exists, the machines can be hard coded to call scan and capture functions. Creating tooling and coding manufacturing machines is an opportunity for the Mechanical Engineers.

Additional functionality from the capture and scan is the second opportunity for future work. Now that we have images and point-cloud models, more computer science students could develop a method for taking measurements off the images and scans. These measurements should be compared to user-defined nominal measurements, and the results can be communicated to the machine for adjustments to be made.

Conclusion

We have created a software package that accomplishes the following functionality:
Image set capture & 3D Scan

Our method was to build an interface that communicates with an Arduino and a USB hub plugged with cameras. The functions can be called with a manual button press, as well as signals sent to the Arduino thereby opening the functions up to automation. In order to prove concept, we built some temporary version of tooling to hold the cameras and laser in place. We can capture images from 5 sides of the table, and scan from the top down.

A future Engineering group may spend time designing real tooling to hold the cameras, that the tool head can pick up and set down. Then they could program the machine to pick up the cameras and call our functions.

A future Computer Science group can develop the images and scans, take measurements, and communicate desired adjustments.

Automation of our functions can result in huge efficiency increases in the machine shop. The inspection process that typically requires a person, time, and separate large pieces of equipment can be eliminated.

References

Thingiverse.com, abarry. (2010, August 28). *MakerScanner v0.3 by abarry*. Thingiverse. Retrieved November 9, 2022, from https://www.thingiverse.com/thing:3946

Barry, A. (2017, January 16). *Andybarry/makerscanner: Makerscanner software*. GitHub. Retrieved November 9, 2022, from https://github.com/andybarry/makerscanner/

McCauley, M. (n.d.). *AccelStepper library for Arduino*. Accelstepper: Accelstepper Library for Arduino. Retrieved November 9, 2022, from http://www.airspayce.com/mikem/arduino/AccelStepper/index.html

Serial. Serial - Arduino Reference. (2019, June 26). Retrieved November 10, 2022, from https://www.arduino.cc/reference/en/language/functions/communication/serial/

String(). String() - Arduino Reference. (2021, May 24). Retrieved November 10, 2022, from https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/

Tkinter - Python interface to TCL/TK. tkinter - Python interface to Tcl/Tk - Python 3.11.0 documentation. (n.d.). Retrieved November 9, 2022, from https://docs.python.org/3/library/tkinter.html

Liechti, C. (n.d.). *Tools*. Tools - pySerial 3.4 documentation. Retrieved November 9, 2022, from https://pyserial.readthedocs.io/en/latest/tools.html

Pickle - Python object serialization. pickle - Python object serialization - Python 3.11.0 documentation. (n.d.). Retrieved November 9, 2022, from https://docs.python.org/3/library/pickle.html

Heinisuo, O.-P. (n.d.). *Opencv-python*. PyPI. Retrieved November 9, 2022, from https://pypi.org/project/opencv-python/

Lundh, F. (2011). *Pillow*. Pillow (PIL Fork) 9.3.0 documentation. Retrieved November 9, 2022, from https://pillow.readthedocs.io/en/stable/

Requeijo, T. (2021, January 30). *Python-configuration*. PyPI. Retrieved November 9, 2022, from https://pypi.org/project/python-configuration/

The python standard library. The Python Standard Library - Python 3.11.0 documentation. (n.d.). Retrieved November 9, 2022, from https://docs.python.org/3/library/index.html

OS - miscellaneous operating system interfaces. os - Miscellaneous operating system interfaces - Python 3.11.0 documentation. (n.d.). Retrieved November 9, 2022, from https://docs.python.org/3/library/os.html

McMahon, M. (2018). *Pywinauto How to's*. How To's - pywinauto 0.6.8 documentation. Retrieved November 9, 2022, from https://pywinauto.readthedocs.io/en/latest/HowTo.html

Pyserial. PyPI. (2020). Retrieved November 16, 2022, from https://pypi.org/project/pyserial/

Appendices

Progress Reports

Development Costs

Progress Reports

CS 4910 – Software System Development and Design II: Implementation and Testing
Project Progress Report

Project Information:
Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 9/19/2022

Team Activity Report:
What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We have two weekly meetings scheduled, one with the client and one without.
2. We have obtained the cameras
3. We have issues and milestones setup in Github, with assignees
4. We established and started a User Interface direction – tkinter
5. We have starting code for the Arduino scanner
6. We have made progress on connecting cameras, with a brute force
7. Makerscanner still doesn't compile. Progress has been made but still not complete.

Client Interaction Report:
Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?
Weekly meeting scheduled for 17:00 Mondays. This week, he showed us access to a tool bench and a plethora of spare parts options.
We decided what coverage he wants: all 5 faces, and we decided the simplest way to do that is with 5 separate cameras. We got our hands on the cameras and a USB hub.
Scanner needs to be widened, likely with multiple scans from different positions. We may need to find a solution for this.

Milestone Review:
Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

We are developing the GUI, camera capture and testing suites for them. We need to get makerscanner to work, and a frame built for the hardware.
We feel on schedule. We have estimated 10 weeks left before the presentation. We want to leave 2 weeks for additional features and improvements, so we should aim to have a completed project in 8 weeks. If we can manage to get the frame built and the cameras to connect to the interface by 9/30, we will have about five and a half weeks to focus on the scanner code together.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

1. Connect cameras to interface
2. Connect scanner to interface
3. Build frame with cameras
4. Develop test suite

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

1. Makerscanner needs to compile
2. If we must make two scans from different positions and mesh them together, this could be a real obstacle.

In effort to prepare for these issues, we will need to make sure we have time. We need to get to a position where we can figure this out quickly so we have time for struggling with this.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 9/26/2022

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We met with client on Monday.
    a. Meeting was brief. Primarily discussed the brick wall we face with maker scanner.
    b. Zach identified a critical weakness with maker scanner, which in short limited our potential image quality to an unacceptable level.
2. Demonstrated new GUI features to client, positive impression.
3. Team met on Wednesday, client joined for some time
    a. Discussed frame concerns
    b. Discussed potential scanner option Ciclop

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

We discussed setting the signals between Arduino and machine
    Scan and capture must be recognized as separate functions
Discussed possibly getting funding for better cameras. 4K.
Pointed out a specific machine we should test on in the lab.
Specific request: Figure out if it's possible to control the camera capture function remotely?

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

We currently have four primary objectives:

1. Build a frame
    a. Bought clamps for cameras
    b. Redesigned base to fit around the specified machine
2. GUI
    a. 'Loading' icon implanted

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

Regarding the frame, some custom pieces must be designed and 3d-printed to hold the cameras and maintain fixed camera positions.
Regarding the GUI, there is now a feature that will identify cameras and title them appropriately. There is also a “dictionary” to maintain states between uses.
Setup Arduino to manage communication for scans/snapshots.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

It has been decided that makerscanner is not the way to go. We must find a piece of successful open-source code to scan objects to meet our requirements. Currently testing the capabilities of Ciclop.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 10/03/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We met with client on Monday.
2. GUI and Camera capture software is running reliably
3. Makerscanner compiles
4. Frame designed and constructed around recently specified machine.

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

Client sees the progress. We discussed a few things:
Look into remote phone camera operation. Potential photo quality improvement.
Signals separate for scan and capture.
Client specified which machine he wanted this to test on, the one in the lab.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

Now that the makerscanner compiles, we need to make it run so we can take a linear scan and create a model.
The GUI is in a comfortable state, waiting for 3D scan implementation.
The frame to hold the cameras is built around the specified machine.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

The frame needs to hold the slider piece.
Actively working on serial connection between Arduino and computer.
Makerscanner compiles but does not run. There is a memory allocation error.
Laserscanner is built around parallel ports, which needs to be updated.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

3D scanning program is still the top priority. We need to create models from a linear scan. We gave ourselves time for this, but not forever.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 11/07/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We met with client on Monday.
2. Class was cancelled so we didn't present
3. Demonstrated connection between interface and Arduino
4. Scanner works
5. Discussed bugs with buttons
6. Discussed making wood piece to hold laser

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

We showed him our connection between the interface and Arduino and makerscanner via bots.
He will not be able to provide any better cameras
Discussed building a piece to hold the laser
Discussed getting money to buy cameras

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

We are developing our final project documents and presentation while we wrap up our piece of software. We are right on schedule. We still have to build a piece of hardware and some code to finish but we have the hard part out of the way.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

We have bots bridging the software together
Ethics report is partially finished
Working out the details between the communication of Arduino and Interface

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

Now that we can control the scanning program, we should be able to create models. Our next objective is to get repeatable and consistent models from the software scan function.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 10/17/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We met with client on Monday.
2. GUI and Camera capture software is running reliably
3. Makerscanner is almost running, progress is made
4. Frame designed and constructed around recently specified machine.
5. Slideshow made, presentation practiced
6. Ethics of respondus completed

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

Client sees the progress. We discussed a few things:
Discussed how to control stepper motors without overriding machine communications.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

Now that the makerscanner compiles, we need to make it run so we can take a linear scan and create a model.
The GUI is in a comfortable state, waiting for 3D scan implementation.
The frame to hold the cameras is built around the specified machine
Need scan function from maker scanner

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

Considering having a bot run makerscanner through our GUI.
Presentation and documents need to be worked on

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

3D scanning program is still the top priority. We need to create models from a linear scan. We gave ourselves time for this, but not forever.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 10/24/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We met with client on Monday.
2. GUI and Camera capture software is running reliably
3. Makerscanner is almost running, progress is made
4. Frame designed and constructed around recently specified machine.
5. Slideshow made, presentation practiced
6. Ethics of respondus completed

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

Client sees the progress. We discussed a few things:
Discussed how to control stepper motors without overriding machine communications.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

Now that the makerscanner compiles, we need to make it run so we can take a linear scan and create a model.
The GUI is in a comfortable state, waiting for 3D scan implementation.
The frame to hold the cameras is built around the specified machine
Need scan function from maker scanner

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

Considering having a bot run makerscanner through our GUI.
Presentation and documents need to be worked on

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

3D scanning program is still the top priority. We need to create models from a linear scan. We gave ourselves time for this, but not forever.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 10/31/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. We met with client on Monday.
2. GUI and Camera capture software is running reliably
3. Bot created to operate the maker scanner interface automatically
4. Camera mounts and frames are ready for cameras to be attached

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

Zach fixed a power supply for Dr. Pavel, and we discussed some details of the project. We talked about power supply and motor controlling.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

Now that we can operate the scanning function via maker scanner, we can start planning how we need the laser, camera, and motors positioned to get best results.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

We have a baseline powerpoint created. The ethics document is also near completion.
We need to attach laser to stepper motor.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

The next roadblock we may run into is the quality of results. We won't know until we are testing how accurate our point cloud models will be.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 11/07/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. Worked on ethics report
2. Zach got the power supply functional and laser is controlled on/off by Arduino
3. We have scans and models!

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?
Client was unable to attend this week.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?
We are continuing to develop our presentation and final document. Going through the document is helping guide us towards completion of the project by tying up all loose ends. We are on track for timely completion of our scope and we may have the opportunity to pursue an additional objective.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.
We have issues set for each section of the final report.
We have successful automatic scans now happening.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?
We are able to get readable models this week. We are having troubles with noisy scans. We should add an objective to manage that.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn.
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 11/14/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

1. Power point is improved
2. Sections of the final document are slowly pouring in
3. Power control for laser is functional
4.

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

We discussed controlling XY stepper motors with lines drawn on a touch screen window. He sees how we have better scans. We talked about the background for his table should be white.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

Zach is soldering together the circuit board for the laser.
We need to attach the laser and camera to the router toolhead.
Interface is being optimized.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

We are continuing to work on final documents. The software is effectively complete.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

We need to complete our documents so we can review and turn them in early for a free review by the professor.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 11/21/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.
Laser power supply, camera/laser mount, UI optimizations, final document pieces

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?
Client is excited we have models and is looking forward to having it on the machine.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?
Zach is soldering together the circuit board for the laser.
We need to attach the laser and camera to the router toolhead.
Interface is being optimized.
We had a practice presentation online in class. It went well for being online. Looking forward to in-person presentations.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.
We are continuing to work on final documents. The software is effectively complete.
Adding ability to configure laser variables from interface.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?
We need to complete our documents so we can review and turn them in early for a free review by the professor. Our presentation is scheduled for 12/6, which is three weeks from now.

CS 4910 – Software System Development and Design II:
Implementation and Testing
Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn
Client: Dr Pavel Ikonomov
Advisor: Mr Jason Johnson
Report Date: 11/28/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.
We have a wooden mount for the laser and top camera that attaches to the router.
We had a few bugs that appeared when testing on different computers, so we had a meeting Tuesday to work on them together.

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?
Client helped provide four bolts for the scan mount.
Discussed the signals being sent between CNCs and our system. In and Out for the scan and image, sent to relay
Client wants to preview our presentation.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?
Power controller for the laser is completed
Frame and mounts are completed
Interface has new bugs
We need to understand what extra libraries need to be installed and automate that.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.
We are continuing to work on final documents and the presentation. We have it broken into sections and assigned to teammates.
Adding ability to configure laser variables from interface.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?
We need to complete our documents so we can review and turn them in early for a free review by the professor. Our presentation is scheduled for 12/6, which is in about 2 weeks as of writing this.

CS 4910 – Software System Development and Design II: Implementation and Testing

Project Progress Report

Project Information:

Team Members: Jaden Perrine, Josh Getsinger, Zach Zeitler, William Sinn

Client: Dr Pavel Ikonomov

Advisor: Mr Jason Johnson

Report Date: 12/05/22

Team Activity Report:

What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.

We have spent time on our presentation and final documents. We brushed out the bugs and have a plan for completion.

We are practicing presentation with client on 12/5, presentation is actually 10:30 on Tuesday.

Client Interaction Report:

Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?

Client has provided a lot of feedback on our presentation. He is supportive to help us complete this.

Milestone Review:

Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?

Power controller for the laser is completed

Frame and mounts are completed

Document draft 1 is complete for an early review with Mr Johnson.

Review with Mr Johnson happened, we got plenty of feedback to improve our paper to get an A.

Issues (or stories):

What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.

We are working on getting this to work on other machines.

We are working on getting videos and pictures for the document and presentation.

Problems and Risks:

What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?

We are running into a problem where the software isn't working the same from computer to computer.

Development Costs:

Camera Mounts:

SLOW DOLPHIN Camera Clamp Mount Monitor Mount Bracket Super Clamp w/1/4 and 3/8 Thread with Cool Double Ballhead Arm Adapter Bottom Clamp for for DSLR Camera/Field Monitor/LED (2 PCS)

Sold by: SLOW DOLPHIN | Product question? Ask Seller

Return window closed on Nov 5, 2022

$20.99

Condition: New

Buy it again

USB Hub:

Targus 7 Port USB 2.0 Hub

Visit the Targus Store

-14% $42.93

Brand Targus

Color Black

Hardware Interface USB

Item Dimensions LxWxH 0.6 x 3.0 x 1.8 inches

Number of Ports 7

About this item

- The Targus 7-Port USB2. 0 Hub with 5V3A PS [Black]

$42.93

FREE delivery December 2 - 8. Details

Or fastest delivery Thursday, December 1. Details

Only 20 left in stock - order soon

Qty: 1

Add to Cart

Buy Now