Automated Production Line For Additive Manufacturing

Ytsen Boersma, Industrial Design Engineering, University of Twente, Netherlands

The idea behind this bachelor assignment is to create a part of a small-scale, automated, and customizable production. Creating part of such a production line can be used to show companies that visit the VRSI Lab, what's possible with additive manufacturing and subtractive manufacturing combined with robotics, and how these companies could implement these technologies themselves. Furthermore, it can give companies an idea of what the impact could be when they swap or move machines within the production line. Having an automated production line with both additive and subtractive manufacturing in combination with AGVs, allows companies to offer more individualized and customizable products, which is the current trend in manufacturing (Kanama, 2017).

For this assignment, the scope will be on realizing the first part of this production line: wireless communication between a 3D printer and an AGV (where the 3D printer sends a signal to a server), followed by the AGV automatically moving to the 3D printer and removing the build plate from the 3D printer. This build plate is then transported to another location, after which the AGV moves back to its starting position.

Based on this assignment scope, the main research question for the project is:

How to make an AGV with robotic arm automatically move to a 3D printer (based on a signal coming from the 3D printer) and remove the build plate from this 3D printer?

To answer this research question, the project has been split in four different topics. The first topic is about creating wireless communication between a 3D printer and a database, where the 3D printer can communicate its status to the database. The second topic focuses on the robotic arm gripper design and 3D printer build plate modifications that will allow the AGV to remove the build plate. The third topic focuses on how to achieve communication between the AGV and the database, which the AGV can use for the automation process. Finally, there is a topic that focuses on the programming of the AGV and making it perform all the required steps automatically.

Status recognition and communication between the 3D printer and database have been realized by using a NodeMCU ESP32 development board. This is connected to a motherboard pin of the 3D printer. During the 3D printing process, this pin's state can be changed with G-code commands (M42: Switch I/O pin, 2010). This allows the 3D printer to trigger a signal when the print is nearly done and a second signal when the print is finished. The NodeMCU ESP32 communicates the status using PHP scripts, which are stored on the same server that hosts the database. These scripts can edit the status values on the database.

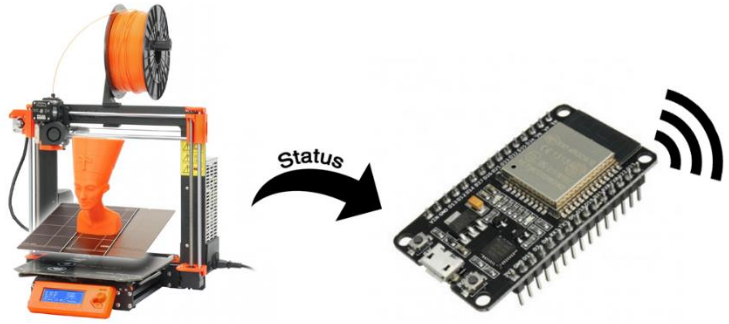

Figure 1: Overview of how the 3D printer status signal is communicated with the server

For the gripper design, an adapter plate has been made to rotate the gripper assembly 90 degrees. This rotation allows the build plate to be removed without adding any gripper points that would interfere with the movement of the 3D printer. Furthermore, a custom build plate has been made that is lighter and has less magnetic force compared to the stock build plate. Finally, a grippy texture has been applied to both the build plate and the gripper jaws. This ensures that the build plate cannot move while it is being transported.

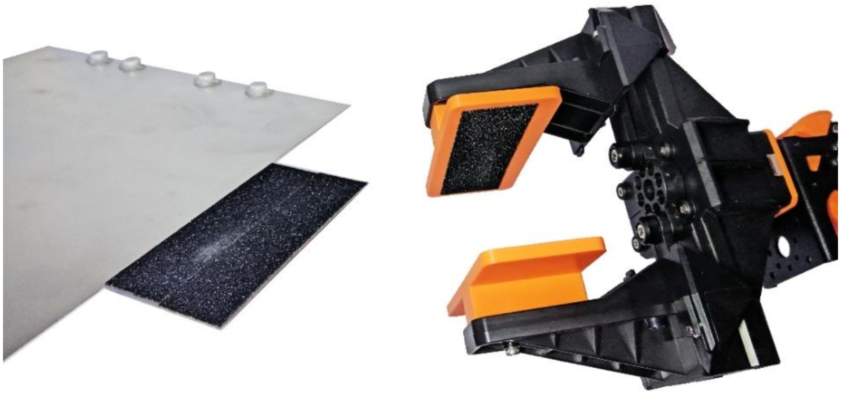

Figure 2: Custom build plate and robotic arm modifications

Communication between the AGV and the database has been realized with a Python script. This script contains the necessary modules and code to directly communicate with the database. The communication code has been embedded inside the script that handles the AGV automation steps.

The specific AGV for this project is the ROBOTIS TurtleBot3, which runs on ROS. This is a framework for robotic software packages that can perform certain tasks (Open Robotics, n.d.). Automating the AGV is done through a Python script. This script communicates with the database and uses the real-time status of both the 3D printer and AGV to determine when tasks can be executed. When a task must be performed, the script will send a message to the AGV manager, which is a ROS package that can control the AGV. For example, when the AGV needs to drive to the 3D printer, the Python script sends the navigation command to the AGV manager, which will then control the wheels of the AGV to navigate it to the 3D printer.

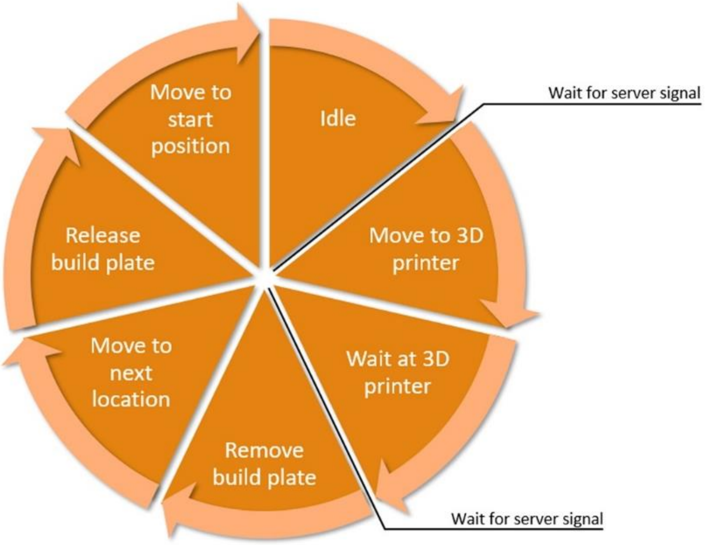

Figure 3: Steps of the AGV automation process

Even though the project has been successfully completed, there are still some aspects that require further research. These aspects include the weight of the build plate and the robotic arm servo torque, how the AGV controller has to be modified when expanding the production line setup, reselecting a development board, researching the AGV battery life and how the AGVs can be charged within the automated production line, and the accuracy and precision of the AGV and the robotic arm.

References

Kanama, D. (2017, December 20). *Manufacturing Transformation toward Mass Customization and Personalization in the Traditional Food Industry.* Retrieved from IntechOpen: https://www.intechopen.com/books/digital-transformation-in-smart-manufacturing/manufacturing-transformation-toward-mass-customization-and-personalization-in-the-traditional-food-i

M42: Switch I/O pin. (2010, June 25). Retrieved from G-code: https://www.reprap.org/wiki/G-code#M42:_Switch_I.2FO_pin

Open Robotics. (n.d.). *About ROS*. Retrieved from ROS: https://www.ros.org/about-ros/