APDS: The Automated Product Delivery System

By

Jason Plew, Dr. Eric M. Schwartz

Jason@mil.ufl.edu

Machine Intelligence Laboratory

Department of Electrical and Computer Engineering

University of Florida

Gainesville, FL 32611

Abstract:

The Automated Product Delivery System (APDS) was developed to address a need in industrial manufacturing for a simple way to transfer materials between areas of a manufacturing environment. At the moment, most equipment used for such a task consist of some type of fixed system, such as conveyor belts, elevators, etc. While adequate, they are not easily adaptable to any changes in their environment. Therefore, an automated unit would seem to be more successful in such an application.

APDS consists of

1. A automated unit that includes a three degrees of freedom robotic arm

2. A communications system that allows a user to send instructions to the robot through any PC

Presently, the user can transmit instructions for the robot to go to two distinct workstations. At the first workstation, it retrieves an object, and at the second, the robot delivers this object.

Introduction:

The user is given a simple program that asks for two workstation numbers. This information is then sent out through the PC's serial communications port, and through an IR transmitter, for the robot to detect.

The robotic unit can be divided into two systems, each controlled by distinct Motorola HC11[1] microcomputers. The base unit is controlled by a standard EVBU[2], mated with a Novasoft ME11[3]. The computer controls the drive motors, as well as two sensor systems.

One sensor system is the IR receiver side of the communications link between the robot and the user. The robot detects the transmission, which then goes through both a filter and software analysis.

The other sensors are a line following system that enables the robot to determine where it is in its environment, and also what workstation it is at.

The other half of the unit is the robotic arm, mounted at the top of the drive section. It consists of three servo motors, each controlling revolute joints: a waist joint, a shoulder joint, and a elbow joint. A wrist joint is not employed at this time. Mounted at the end of the forearm segment is an IR sensor, allowing the robot to detect the object being retrieved.

The robotic arm is controlled by the Novasoft MPR11[4] microcomputer, and is also mated with an ME11 board. The EVBU and the MPR11 are independent, though they do communicate at certain points in the process. Once the robot is at the right workstation, the EVBU toggles an output. The MPR11 detects this transition, and directs the robotic arm to either pick up or drop off the object. Once it accomplishes this task, a similar technique is used to inform the EVBU that the arm is done, and the base unit continues on its way.

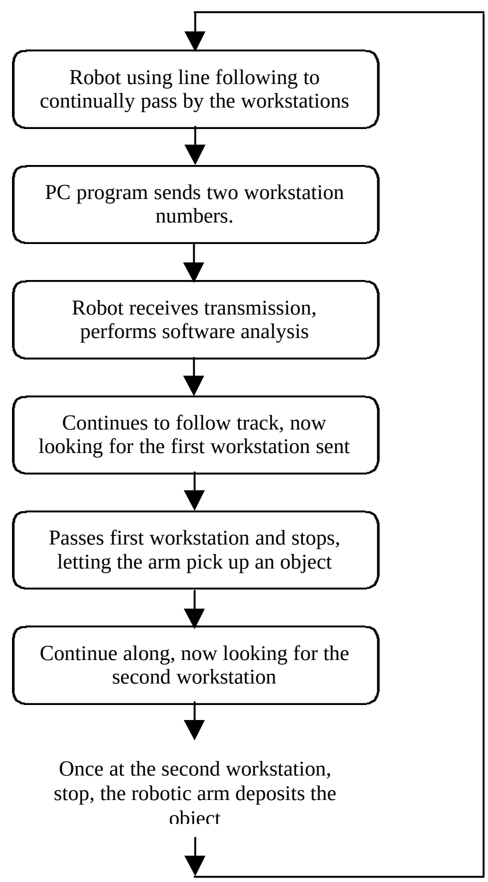

Figure 1: System Flow

Both microcomputers were programmed in Newton Lab's Interactive C (IC) v2.85.[5] New software drivers were developed for the system in the IC-Binary format.

The Base Unit

The base unit consists of a EVBU/ME11 microcomputer controlling the drive motors, the Line Following (LF) sensors, and the receiver side of the communications link. The software consists of a simple main() function that initializes certain variables, and then shuts off after executing the software for the sensors. Utilizing IC's multitasking abilities, separate process's are started for the Communications software and LF sensors. They do share data on which workstation is being passed, as well as the instructions from the user. Both processes can also control the drive motors. However, they also have the unique ability of being able to turn each other off at certain points in the process. This will be explained further later on.

The base unit's body looks something like a tank, with the LF sensors mounted at the bottom of the front edge, and the receiver system in the back at the top. The base unit has three separate internal bays, with the microcomputer in the back, drive motors in the middle, and the LF interface (along with the robotic arm electronics) in the front. (See Figure 6). Actual treads were originally used, but have been replaced with regular wheels.

The Line Following and Barcode Reading Sensor System

This sensor system consist of four Hamamatsu P306201s. Normally used a shaft encoders, in this case they have been used solely as an integrated IR emitter detector package.[6] The IR light is either reflected by the light surface of the floor, or absorbed by the black tape.

The two middle sensors are on both sides of the center black strip of tape. (Refer to Figure 2) If either sensor crosses the path (whether due to a turn or to wandering), the robot can determine which direction to turn so that it aligns itself back to having the tape run directly beneath it.

The sensors on the sides work in the same way, but for determining which workstation the robot is passing. The strips on the left together make up a barcode, representing the workstation coming up. The left sensor counts how many strips of tape it detects. Once the right sensor detects a strip, called the stop tape, the robot assumes that it has passed all of the barcode, and that the number of strips the left sensor has seen references the workstation next to it. If that workstation matches the one whose number had been transmitted by the user, the robot will stop, and the arm will perform its task. Either way, the robot will continue along, starting over in the process of looking for the barcode.

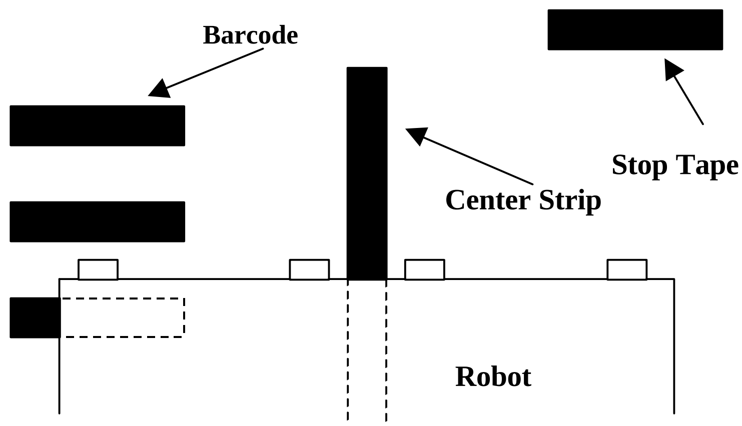

Figure 2: LF Sensors and Environment Layout

The Communications Link

The communications link consists of a IR transmitter based at a PC, and an IR detector on the robot. A program was developed in ANSI standard C that prompts a user for a workstation that has an object to be retrieved, and another workstation where that object is to be dropped off. These two numbers are then converted to binary, and a 4-bit "synch word" is inserted at the front of the data. At this present time, only 4 workstations are supported.

This now 12-bit word is then continuously sent out the computers com port. A simple circuit utilizing a 555 timer modulates the signal at 40 kHz. Finally, the signal is sent out through a IR emitter. At the present time this

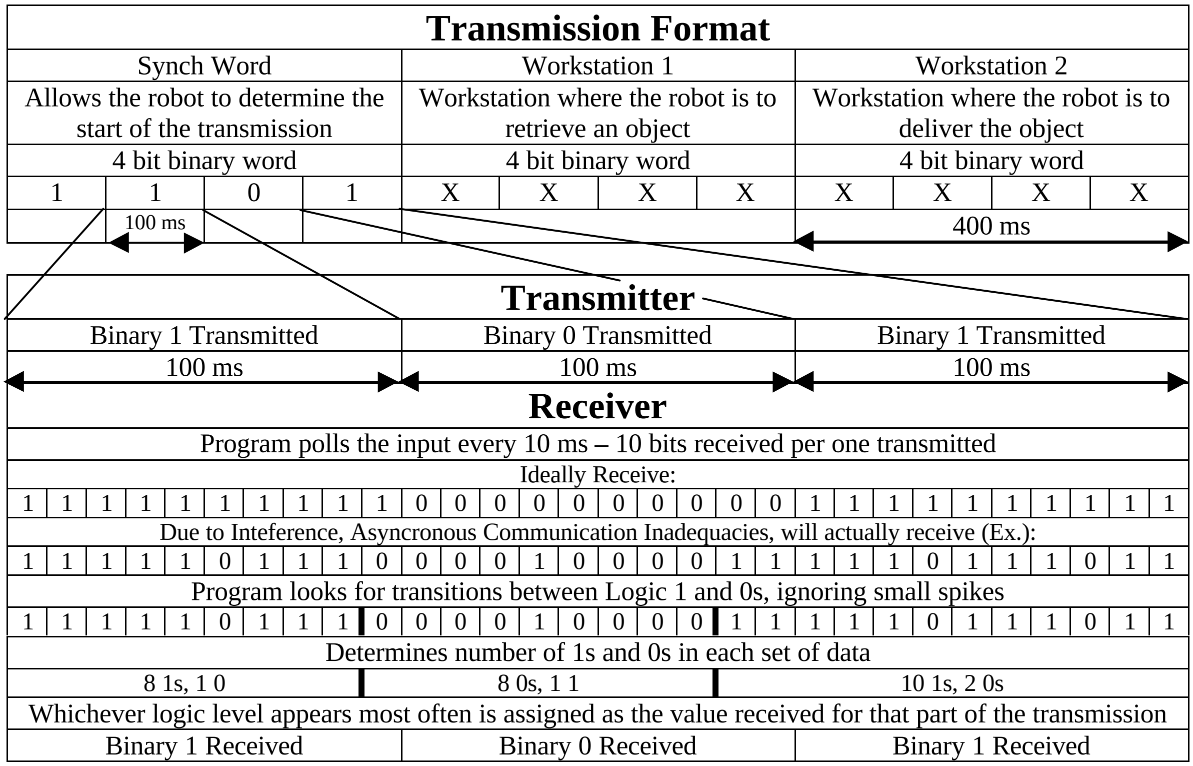

Figure 3: Communications System Transmission and Analysis Overview

emitter is simply a TV remote control, which has been modified so that the inputted signal determines the transmission instead of the normal memory chip containing TV control codes.

The robot's program, meanwhile is allowing the line following software to control its actions. The communications software is in a continuos loop, waiting to continue until it detects a transmission. Once that signal is received, the line following software is shut down, and the robot stopped, so that the full attention of the computer is devoted to receiving and decoding the transmission, and interrupts from other devices will not conflict with the communications program.

The robot "sees" the transmission using the Sharp GP1U58Y IR detector. The signal is demodulated by the sensor, and then goes through a low pass filter, as well as a software analysis that performs its own filtering process. This removes the small interference that can be introduced into the transmission from nearby light sources. Though the filter was not part of the original design, it was found that in certain environments, the interference would cause the transmission to be so distorted that the software analysis would be unable to compensate. Since the filter was added, there have been no further problems.

The software analysis forms the bulk of the communications software. Each bit transmitted lasts for 100 ms. Therefore, the robot polls the output of the receiver every 10 ms, theoretically collecting 10 bits of data for every one sent. The software polls the receiver for 2.5 seconds, storing 250 bits of data. Theoretically, the transmission should cycle twice during this time. The software looks for the first transition. It then goes through the data, ignoring small changes in the signal between a high and low level (assuming interference), and taking all the data between the transitions (which should be close to 10 bits), and assigning a 1 or 0 to the value of the transmission at that time, depending on which value appeared more often. The end result is 24 bits of data, or the 3 word transmission sent twice. Refer to Figure 3.

The robot then looks for the synch word that had been originally inserted at the beginning of the data. Once it has been located, the next 4 bits are converted back to decimal, and assigned as the workstation number where an object is to be picked up. The same is done to the final four bits, representing the workstation where the object is to be dropped off.

Finally, the Communications software restarts the LF subroutine, passing along the two workstations it should be looking for. It then shuts itself down, not to be reactivated by the LF software until after the robot has dropped of the object at the second workstation.

The Robotic Arm

The robotic arm is the other half of the automated unit. Completely separate, it has its own software, run by its own microcomputer located in the front internal bay. The software consists of drivers to control the servo motors of the robotic arm, and a program that directs the motors in a certain path, along with looking for the object with an IR sensor, and communicating with the EVBU.

The robotic arm itself has three DOF. It is similar to a Articulated manipulator, but with a fixed wrist. There is also no true end effector, though two prongs extend from the end of the forearm segment that can "scoop" up the object. (See Figure 4 and Figure 5).

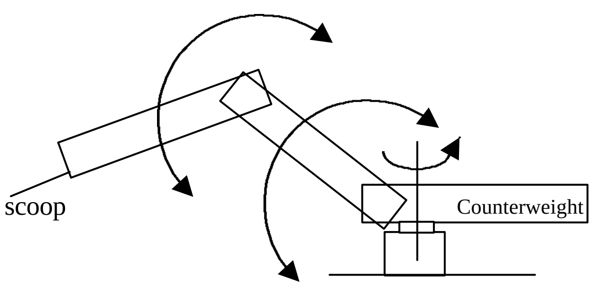

Figure 4: Robotic Arm Design

The movement that robotic arm goes through to pick up the object was basically preprogrammed. The object is assumed to be at the very edge of the workspace, because of the nature of the "scoop" end effector. The only variable is the rotation the robot must move to align itself with the object. For the program, the object was placed at the edge, and the position of the servos were slowly changed until the object had been picked up. These movements were then combined into the program so that the robot moved at a regular speed, yet in the exact same manner. Finally, the exact opposite of these movements were also placed in the program, for when the robot is setting the object down.

While it is possible for the object to be further in, eventually the robot will be unable to scoop up the object. The addition of a true end effector, however, will improve the volume of the robotic arm's workspace.

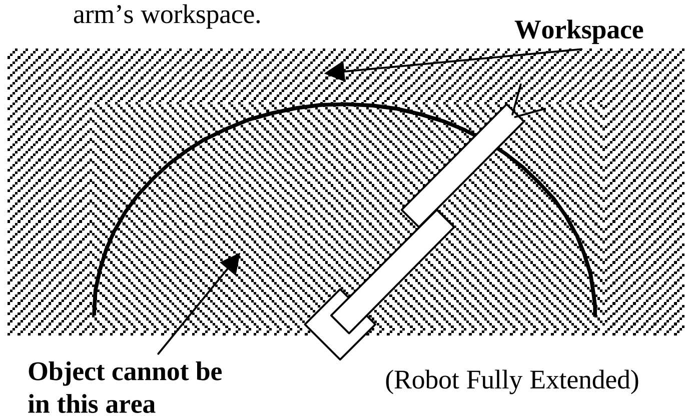

Figure 5: Robotic Arm Workspace

Future Development

Several improvements are intended for the Automated Product Delivery System. First and foremost is to correct several mechanical issues. The means of locomotion will have to be improved, as the present wheels mounted on the drive motors are slipping, resulting in the robot constantly being unable to turn correctly, thus interfering with the process of line following. Also, the batteries that power the servos will have to be replaced, as the current ones do not supply enough power. The servos themselves may also be replaced with motors that are more durable and have a greater torque.

Other improvements include adding a gripper system to the robotic arm, and developing some means to allow an object to be retrieved no matter where it is located within the workspace of the arm, as mentioned earlier. The communications system will be modified to allow more then 4 different workstations. To accomplish this, a longer and more unique synch wold will be used, and the length of the transmission will be increased.

Finally, the system itself will be modified in some manner so that the robot will not have to rely on line following techniques to find its way through the environment. This new sensor system, undetermined at the present time, will allow the robot to find a specific workstation, without modifications to the area having to be made prior to the system being operated.

References

[1] HC11- M68HC11 Reference Manual. Phoenix: Motorola Literature Distribution, 1991.

[2] M68HC11EVBU - Universal Evaluation Board User's Manual. Phoenix: Motorola Literature Distribution, 1992.

[3] Mekatronix ME11 Expansion Board for the MC68HC11 EVBU, Assembly Manual, Gainesville, FL: Mekatronix, 1997

[4] Mekatronix MPR11 Microcomputer, Assembly Manual, Gainesville, FL: Mekatronix, 1997

[5] Martin, Fred, The 6.270 Agent Builder's Guide, The Epistemology and Learning Group, MIT, 1992.

[6] A M. Flynn and J. L. Jones, Mobile Robots - Inspiration to Implementation. Wellesley, Mass: A.K. Peters, 1993.

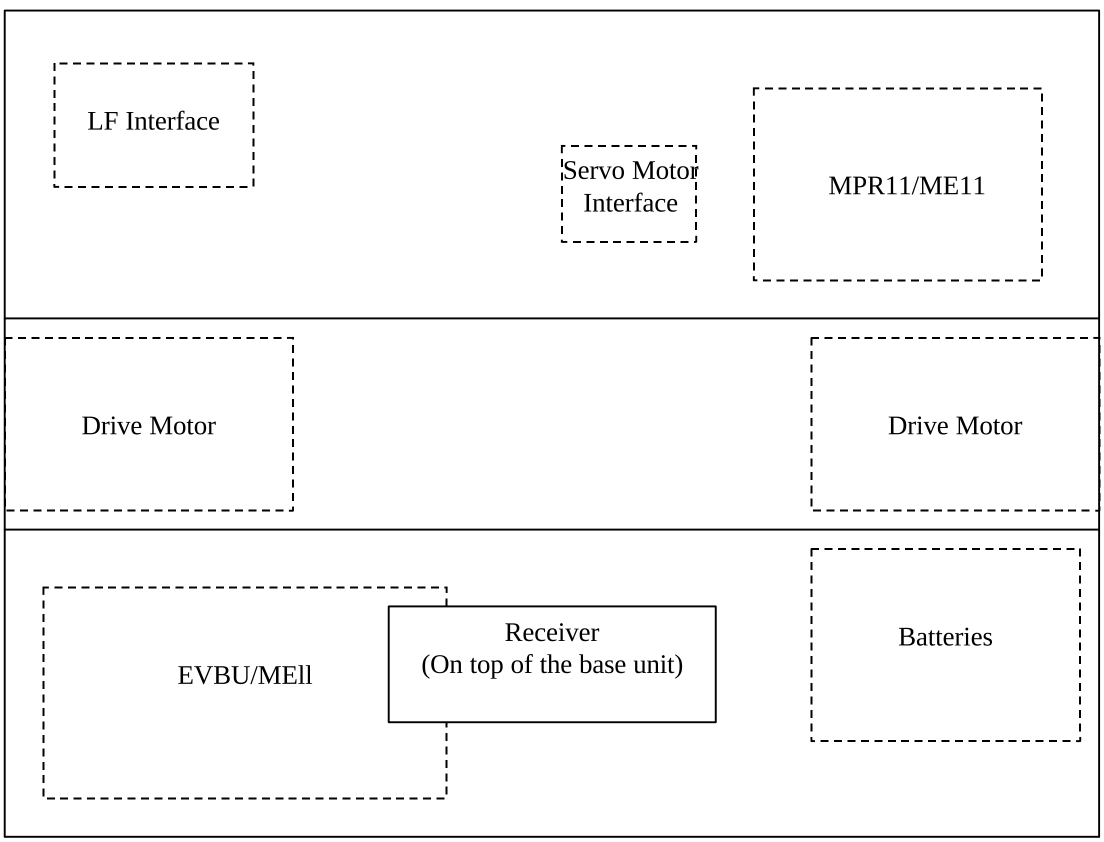

Figure 6: Robot Layout