ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Implementation of Pick and Place Operation, Battery Backup and Zone Control in Automated Guided Vehicle

Saravana Sundar.T[1], R.C.Vignesh[2], G.Moorthi[3], Nivedhan.M[4]

Final Year, Kongu Engineering College, Tamilnadu, India [1, 2, 3, 4]

ABSTRACT: High demands on manufacturers have left their shipping warehouses in havoc. Human error has a negative effect on safety, efficiency, and quality. These expenses are reduced with the introduction of an Automated Guided Vehicle, AGV. A driverless, intelligent forklift uses an optical path to quickly and safely traverse a warehouse. Its capabilities are enhanced by the ability to send and receive tasks through RF data communication.

I. INTRODUCTION

Advancements in manufacturing technology allow companies to rapidly produce products. This has provoked a trend to reduce bulk inventory in favor of short term supplies. Although this allows corporations more financial freedom, it requires warehouses to accommodate temporary, selective storage. Improved product handling and speed can be achieved with the implementation of an Automated Guided Vehicle, AGV. In a traditional warehouse, human safety governs the productivity. With the help of intelligent computers, the AGV can safely achieve higher speeds. Precision turning allows it to accurately navigate in tight spaces. The AGV is highly flexible as a result of remote communication. Its ability to communicate with other autonomous vehicles provides a seamless operation. Continuous coordination between vehicles delivers money saving efficiency. The introduction of unmanned vehicles onto a warehouse floor has favorable effects on safety. With the aid of environmental sensors, the AGV can detect objects in its collision path. Automation eliminates vehicle traffic jams and their potential for accidents. For companies building new warehouses, there are many monetary benefits to investing in intelligent machinery. The workforce required to run the warehouse and the additional overhead (e.g., insurance) required to support that overhead will be drastically reduced. Increased product turn-around and faster shipping will result in more satisfied customers. Also, automation reduces the risks of personal injury.

II. INTEGRATED SYSTEM

The AGV operated in a model warehouse, built to scale. Its primary task was to relocate pallets within the warehouse. A external input generated by an infrared remote control notified the AGV whether a pallet was entering or exiting the warehouse. To get to its destination, the vehicle traversed the warehouse by following high contrast lines. When the four pair line follower module detected an intersection, the AGV determined whether to turn or go straight by using an algorithm that incorporated the vehicles current location and direction. In the first situation, the remote control signaled that a pallet was entering the warehouse. The AGV picked up the pallet off of the incoming shipping dock and dropped it off at one of several docks at the other end of the warehouse. Through RF communication, the AGV told another autonomous vehicle, the ASRS, the new location of the pallet. The second situation allowed for a pallet to be shipped out of the warehouse. The AGV waited for the ASRS to confirm that it dropped off a pallet at one of the transition docks before it picked it up and moved the pallet to the outgoing shipping dock. While traveling, the AGV polled two forward facing infrared range finding sensors to determine if an object was in its forward collision path. If and object was detected, the vehicle would stop and wait for the obstruction to be cleared. While reversing, bump sensors detected the occurrence of a rear collision, which would permanently disable the vehicle.

Mobile Platform

The designer intended the AGV to be a small vehicle that could quickly traverse a warehouse and move pallets around. Creating the smallest vehicle possible required creative thinking and experimentation. A concept drawing was entered into AutoCAD 2005 and rendered in three dimensions. After the blue print was verified, a T-Tech prototyping machine cut the platform out of eighth inch plywood. The top of the vehicle was constructed from thin balsa wood, which was glued to a wood skeleton that matched the curve of the AGV. Although the fork protruded through a rectangular cutout, the lid could still be easily taken on and off. A second cutout was made for the infrared distance sensors, and the display lens off of a V.C.R. was modified to conceal the sensors. The platform was primed and painted with black and silver lacquer spray paint, and a high gloss was created by applying polyurethane.

The body was five and a half inches wide, five and a half inches long, and six inches tall. A rectangle was cut out of the back and bottom to provide for the housing of the LCD and line follower module, respectively. Arches were cut out to provide clearance for two 2.2 inch diameter tires, which were mounted on the rear end of the robot, while two casters supported the front of the robot. Three tier shelving was used to optimize space inside of the AGV. The center of gravity was intended to be above and forward of the motors. However, an error on the chasis design was quickly discovered once it was assembled. A third caster was added to the extreme rear of the vehicle so the robot wouldn't tip back while driving forward. The height of the new caster was precisely measured so the vehicle would rock as little as possible.

Actuation

The AGV required the capability to move around a model warehouse built out of a sheet of plywood and to pick up pallets once it reached it's destination. Two types of actuation where needed to meet these objectives. Drive motors and tires added the function of movement, while a third motor supported a pallet jack.

Wheels and Movement

The AGV was propelled by two 200 R.P.M. D.C. gear head motors, which where attached to the rear tires. The tires were 2.2 inches in diameter and one inch wide. With the addition of the third caster, the majority of the vehicle's weight rested on the rear caster, and the tires were slipping. Thick coats of rubber cement were painted onto the tires and the vehicle regained traction. The AGV had an excellent turning radius as a result of the platform layout. The tires resided one inch from the middle of the robot, which almost allowed it to turn in place.

Forklift Mechanism

The forklift was created out of wood and mounted to the platform with hinges.

Serendipity struck when the hinges could not be installed perfectly straight. The hinges had extra friction, and the fork did not flop around even though they were still moveable. An arm was rigidly attached to the forks and mounted so that it was parallel to the forks and in the opposite direction. A slot in the arm was created with a T-Tech prototyping machine and connected to a servo arm with a pin. When the servo was in its neutral position (zero degrees), the forks were down and parallel to the ground. Altering the servo's pulse width moved the servo arm back to negative thirty degrees and the forks up to positive thirty degrees. Wooden stops where mounted on the fork to prevent pallets from sliding back and crashing into the robot.

Sensors

Infrared Detector

The goal of this project was to develop a robot that streamlines the warehousing process. An infrared remote control allowed the user to dynamically communicate with the vehicle on the factory floor. Buttons on the remote control corresponded to requests to store or retrieve a pallet. A Sony television remote control (Sony code # 202) was used to send infrared data. Each button on the remote had a unique bit pattern. When a button was pressed, the remote formed a packet

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

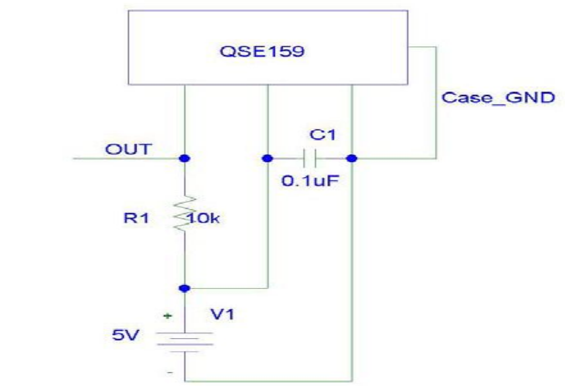
Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

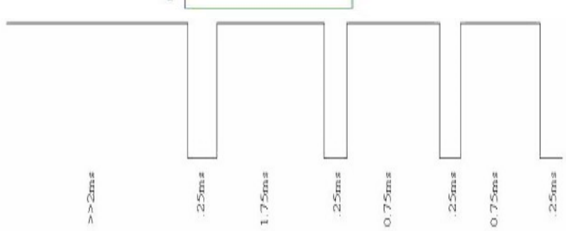
Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

of data including start bits, data elements, and stop bits. Digital Signal Encoding was used, and the packet was sent serially through a 40kHz modulator. Modulating the signal decreased the probability of ambient infrared crosstalk.

After modulation, the signal was sent to the infrared generator. The AGV used a airchild Semiconductor infrared detector to receive the signal (Figure 1). A band pass filter was incorporated into the detector so that only 40kHz signals were accepted. When the sensor detects infrared heat, it demodulates the signal and sets the output pin low. Internally, the sensor used a Schmitt trigger to reduce switching noise on the output pin. This was highly desirable because false pulses could be mistaken as part of the incoming bit stream. The output was connected to the microprocessor's low priority external interrupt, which was configured to a falling edge. In the interrupt subroutine the data was serially converted into bits by analyzing the length of the high pulses. The signal was first analyzed on an oscilloscope. The researcher noticed that all of the pulses (excluding the start pulse) had one of two qualities. There was always a 0.25ms high or low pulse which was followed by a 0.75ms high period (Figure 2.). This made it possible to have either a 1ms high period or a 0.75ms high period. The back end of one 0.75ms pulse plusthe 1ms pulse provided a total pulse width of 1.75ms. This long pulse width never appeared back to back.





In all of the bit patterns, there were several more short pulses that long pulses. The researcher guessed that the data was riding on the long pulses, so he decided to call the short pulses '0' and the long pulses '1' (Figure 3.). The bit patterns were collected for the entire signal. A pattern was discovered upon analyzing the data. After the first five samples were ignored, the subsequent five samples where stored into an array. All other samples where ignored. The array was reversed so that first sample collected became the least significant bit in binary form. The array was converted into a hexadecimal number. With this procedure, button "one" on the remote gave the hexadecimal number of 1, and button "two" on the remote gives the hexadecimal number 2. This was the Sony code!

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

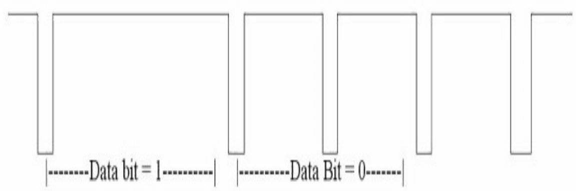
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

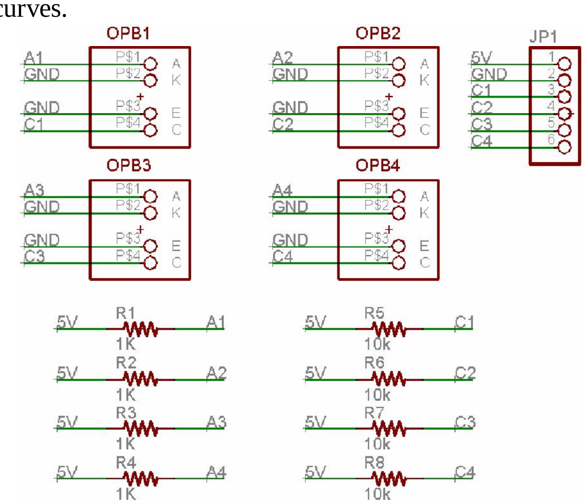


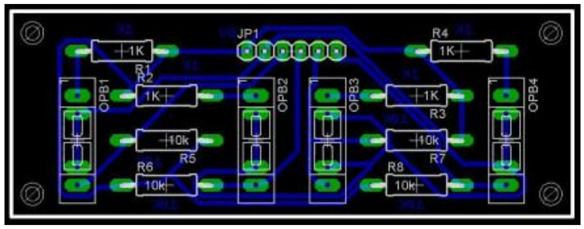
Proximity

It was desirable for the AGV to be capable of safely traversing a warehouse without colliding with obstacles in its path. Two forward facing Sharp GPD2D12 infrared range finders were placed approximately two inches apart. Both where pointed 30 degrees toward the center of the robot. Obstacles were detected when something passed into the sensor's line of sight. The sensors were preassembled and powered up with 3.3 volts. The analog output was connected to the microprocessor. Because the robot was operating in a model warehouse, close range sensors where chosen to stay consistent with scale. The farthest distance the GPD2D12 can measure is 80 centimeters. With a body length of 13 centimeters, the AGV is considered to be a 1:14 scale. Applying the scale, a life size AGV could detect obstacles 35 feet in front of it.

Line Follower

Navigation will be achieved by following black lines on the warehouse floor. A four pair line-tracking module is constructed with Optek OPB745 Reflective Object Sensors. They are constructed with infrared light emitting diodes coupled with phototransistors (Figures 4 and 5). Because the reflective properties of black and white surfaces are different, the sensor will return varying analog values relative to the surface they are above. The microprocessor polls these analog values and converts them to digital data: black is 23016 and
white is 13516.Two sensors are offset one half of a centimeter from the center of the module. This allows the robot to center itself on a two centimeter wide strip of electrical tape. Both of the other sensors are three centimeters from the middle. They serve to detect intersecting black lines. Combining two center sensors with an outside sensor allows the AGV to distinguish intersections from curves.





Collision

Two active low bump sensors are located on the back of the robot. They are wired in parallel, and the output is tied to a low priority interrupt. The interrupt is configured to a falling edge. In the event that the interrupt is fired, the AGV permanently stops.

RF Link

The AGV is designed to work hand to hand with another vehicle on the warehouse floor. Communication is achieved with a Laipac TRF-2.4G RF transceiver. The development of this system was completed by Albert Chung, and it will be inserted into the AGV as a "plug and play" device. A complete Special Sensor Report on the RF data link can be found at http://plaza.ufl.edu/tskipp/agv_asrs/RF.htm . Laipac merged several devices into one convenient package: a bidirectional transmitter, Cyclic Redundancy Check generator, and an antenna. The transceiver uses an external clock to serially input data from a microprocessor. Once the internal data buffer is full, the chip uses Shock Burst® technology to assemble a packet: including an internally calculated preamble and CRC. Data is transmitted with a signaling rate as high as 1Mbps. To address the possibilities of errors, the Stop and Wait Automatic Request protocol was used. This had several advantages over direct communication, including an alternating frame number and positive acknowledgment. If frames were received out of order, the receiver would NAK the sender and wait for the correct frame. However, things were not perfect and the two robots went quickly out of synchronization. To account for this, software allowed for the dynamic resynchronization of frame numbers. The biggest source of error was ambient noise that crosses talked to our system. We implemented header error control to help counter this. By inserting a standard header in the unused bits, the receiver could test the incoming message. Another possibility of error comes from both devices transmitting at the same time. Both robots were programmed with separate timeout lengths.

Thus, if one robot sent something and never received and ACK, it would resend its packet before the other would.

Behaviors

The AGV uses several behaviors to complete its objectives. These behaviors where programmed in separate modules, and an arbitrating function coordinated the events. Priority interrupts were used to address the precise timing requirements of some devices.

Communication

The AGV originally utilized two forms of communications in the forms of an RF transceiver and an infrared remote control. The ASRS required more data because it needed to keep track of the location of pallets, different shelf heights, and pallet ages. With the remote control on the AGV, the AGV simply passed all of the information to the ASRS and then discarded everything it did not need. This created a lot of overhead, so the remote control was moved to the ASRS, which in turn passed the AGV the little information it needed.

The ASRS communicated to the AGV under one of two circumstances. If a pallet was being shipped into the warehouse, the ASRS would immediately tell the AGV that a pallet was incoming and to what dock the AGV needed to deliver it to. Once the AGV picked up the pallet and dropped it off at the correct dock, it was transmit the exact packet back to the ASRS. This would notify the ASRS that the pallet has arrived on the bottom shelf and needs to be moved to a higher shelf, which would clear out the dock for more incoming pallets. The second scenario was the outgoing pallet routine. The ASRS moved a pallet to the bottom shelf and told the AGV two pieces of information: the dock location and that the pallet was outgoing. Immediately after the AGV picked up the pallet, it notified the ASRS that it completed the task. The ASRS was now free to reuse the dock.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Queue

The AGV was much faster than the ASRS, so the delivery and storage of pallets was not one to one. The implementation of a queue insured that the neither of the robots would ever be sitting idly by. Upon the successful reception of an RF packet, the AGV pushed it onto the queue. When there were no jobs being processed, the robot continually polled the queue for a new job. The jobs where handled in a "First In First Out" order.

Tracking

The four pair line tracking module was used to navigate the AGV on a dark brown, glossy floor with white strips of electrical tape. Three motors speeds were defined: medium fast, medium, and slow. Normal navigation was done with the medium fast speed. If the vehicle started to stray off of the line, the software would notice a difference in the values from one of the two center sensors and decrease the appropriate motor's speed to medium. If the vehicle completely left the line, the robot turns in the opposite direction of the last sensor read. For example, if the robot last saw "white" on the right-center sensor, it would turn left. The software detected an intersection when the output from either of the outer line following sensors read "white." The detection of intersections allowed for mapping system on a Cartesian coordinate system. The warehouse was laid out so that the vertical segments of line had Y values of negative one, zero, and one (where negative one was closest to the shelves). The X segments had values ranging from zero to four, which were coordinated with the dock numbers. There was an obvious need for direction when the robot was turning; the robot needed to turn right if it were traveling one away and left if it were traveling another. A cardinal direction system was implemented: zero represented East, one represented North, two represented West, and three represented South. Each time the vehicle turned, the software would adjust the direction.

Lifting

For all jobs, the AGV first picked up a pallet and then dropped it off at a new location. Two white lines were placed in front of each dock and allowed the arbitrating function to make different actions. At the first line, the AGV would lower its fork before driving in. It would raise its fork at the second line before turning around and heading toward its destination. Similar procedures were followed when the AGV went to drop the pallet off.

Obstacle Detection

Although there was no need for humans in the automated warehouse, people could be unpredictable (unlike robots). Distance sensors determined if an object was in the forward path of the vehicle. If they detected something, the robot would pause before checking to see if the obstruction was still present. For obvious reasons, the sensors were temporarily disabled when the AGV was approaching the shelves. While backing up, the rear bump sensors were activated so a rear collision could be detected. If this event occurred, the vehicle would be permanently disabled.

Experimental Layout and Results

Experiments were performed on all of the sensors individually before they were compiled into one integrated system. Tests on the infrared distance sensors yielded analog values that corresponded to their distance from objects. The value was read for six inches and hard coded into the program. The sensor outputs from the infrared line following module varied from each other; typical readings from left to right were 165, 145, 99, and 148. The values were highly conditional of ambient infrared sunlight, so the module was recalibrated to white every time the robot powered up.

Experiments were performed on a Sony television remote control (code number 202) which generated an IR signal and was detected using a Fairchild Semiconductor QSE159. It was desirable to know the precise signal outputted from the infrared detector. Initial tests were conducted on the PIC18F8720 microprocessor, which operates on a 5MHz internal clock. Random data values were collected, and the researcher was unsure whether they were the result of the infrared transmitter, infrared detector, or the software running on the microprocessor. To remedy this solution, the detector was connected to an oscilloscope, and an algorithm was created that characterize the signal. The oscilloscope was also used to measure the lengths of start and data bits. These lengths were used in software, and the LCD screen displayed the pattern when a remote button was pressed. All tests were performed in a small room with the window blinds closed. The room was lit with a fluorescent light bulb.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

IV. CONCLUSION

All deliverables set forth in the project proposal were successfully met. The designer initially intended the AGV to be a super fast vehicle. Even though it could follow lines at high speeds, it occasionally missed intersections. Although the AGV had to be slowed down, it was still much faster than the ASRS, which met specifications. The forklift mechanism was not built when the platform was designed and assembled, and the original concept for the forklift failed. The limited amount of space hindered subsequent ideas for the forklift, and several different designs were prototyped before the final version was built. However, the forklift turned out very well and the wait was well worth it. The biggest area for improvement was in the warehouse. Although it took forty hours to build and cost seventy dollars, there were inherent flaws. First, the ground was not perfectly smooth. Paneling was used on the AGV side while plywood was used on the ASRS side. The plywood surface would not sand down, and was quite bumpy. This jostled the ASRS while it was driving and gave inaccurate depictions to the line follower sensors. Another problem with the warehouse was the warping shelves. The ASRS had a maximum fork height of thirteen inches, so there was not much room left for pallets. Thick wooden shelves would have greatly reduced the clearance for each shelf, so half inch plywood was used instead. Although this problem was overcome in software, metal shelves would have simplified things. The AGV design project went fairly smooth. The most frustrating problems stemmed from Microchip's MPLAB compiler. The project designer was highly skilled in the C language and had to make adjustments to his coding techniques to adjust for inadequacies in the compiler. Although none of the problems were detrimental or prevented something from eventually working, many weekends were wasted on "something stupid." I would recommend future students who enroll in the Intelligent Machine Design Laboratory to talk to someone who has used their compiler. They might be able to save people an ample amount of time by passing words of wisdom. There are many areas for improvement on this project. A Sliding Window Automatic Request protocol could be implemented to further reduce RF transmission errors. Both robots could continually send a null packet of data to each other, which would allow the robots to know if they went out of range. More importantly, it would allow them to see how many errors they were receiving, and they could dynamically adjust for it. A larger warehouse with more shelves would allow the demonstration to be more meaningful.

A swarm approach could be implemented by building several AGVs for every ASRS.

Conveyor belts could lead in and out of the warehouse and notify the vehicles when a part is present. The easiest improvement would be to add an RF link to a personal computer and replace the remote control with graphical software.

V. DOCUMENTATION

Fairchild Semiconductor Datasheet:
http://rocky.digikey.com/scripts/ProductInfo.dll?Site=US&V=46&M=QSE159
LaipacTRF-2.4G Datasheet:ttp://www.sparkfun.com/datasheets/RF/RF-24G_datasheet.pdf
Nordic Semiconductor nRF2401 Datasheet:
http://www.sparkfun.com/datasheets/RF/nRF2401rev1_1.pdf William Dubel's Reliable Line Tracking Report:
http://www.mil.ufl.edu/imdl/handouts/lt.doc