Automatic Transport System For Industrial Applications

Nithin Pradeep

Signalling Design Engineer, Atkins (India) Pvt Ltd., 10th Floor, Safina Towers, Ali Asker Road, Bangalore-560052, India

Abstract: ***Transportation is a serious issue deserving due concern in industries. Raw materials and finished goods must be moved within different work stations in an industry. The overall efficiency and production rate of an industry can be improved drastically by employing microcontroller based robots for carrying out these movements. This paper demonstrates the implementation of the prototype of such a robot. The robot is controlled by an ATMEGA 16 microcontroller. The communication with the robot from an external source is realized through a ST-TX01 315/433.92 MHZ ASK Transmitter and ST-RX04 315/433.92 MHZ ASK Receiver. The data transmitted is encoded through a HT12E chip and the same is decoded with HT12D chip fitted inside the robot. This decoded information dictates the orientation (clockwise or anticlockwise) of the gear motors fitted on the robot. The gear motors are interfaced through the motor driver IC L293D. The robot moves to its destination by tracing a black line (in white background), for this purpose sensors are fixed on the bottom part of the same .The entire behavior, motion sequences etc, that the robot had to follow was written as a program and was burned into the ATMEGA 16 microcontroller. The software used for this was AVR STUDIO (provided by the manufacturer of the ATMEGA microcontroller)***

Keywords: Transportation, microcontroller, embedded C, sensors, wireless data transmission.

1. Introduction

Industries rely on internal transportation to a great extend. Every industry has work stations or sections where a part of the final output is synthesized or modified and these sections might need various accoutrements or raw materials which had to be supplied from the main store house in the industry. The production rate, efficiency and the overall effectiveness of an industry can be enhanced significantly if we are employing robots to facilitate this transportation within the industry. This paper demonstrates the research done to design and implement the prototype of such a robot.

The wireless data transmission technology is utilized to a great extend in the design of this robot [1].A set of inputs are send to the robot through a transmitter via Amplitude shift keying [2].The robot is fitted with a ST-RX04-ASK receiver [3] which receive this information. The transmitter used is ST-TX01 [4], which is an ASK Hybrid transmitter module. The ST-TX01 is designed by the Saw Resonator, with advantages like effective low cost and small size. It has an operating frequency range of 315 to 433.92MHZ. It needs a Supply Voltage in the range of 3 to 12V and has an output Power in the range 4/16dBm [5].The ST-RX04 receiver is an ASK superhetrodyne receiver module [6] with PLL synthesizer and crystal oscillator, it has a receiving frequency in the range 314 to 433.92 MHZ and an operating voltage of 5V and has an intermediate frequency of 500KHZ and a typical sensitivity of 105dBm.

The information which had to be transmitted contains the identity of the work station which called the robot. Hence each workstation should have a transmitter. But providing one transmitter for each work station can increase the cost of the system significantly. Hence we use an encoder to convert the original data which is in parallel form into a serial format so that the data from all work stations can be transmitted from a single transmitter. The encoding is done by the IC HT12E [7].The transmitted data upon reception is then decoded using the IC HT12D [8] at the receiving side. It is capable of decoding 12 bits of information and has 8 address bits and 4 data

Bits. Other features of the HT12E/HT12D ICs are that it has an operating voltage in the range of 2.4 to 12V and employs CMOS technology [9] as a result they have high noise immunity. The receiver is located inside the robot.

The data coming from the decoder is given to the microcontroller. The microcontroller is responsible for controlling the robot. In this prototype I have used the ATMEGA16 [10] microcontroller chip manufactured by the ATMEL corporation. ATMEGA16 is a high performance microcontroller chip made with advanced RISC architecture [11] and possesses high endurance non volatile memory system (16 Kilobytes of in-system self-programmable flash program memory, 512 bytes EEPROM and 1 Kilobyte internal SRAM) and also have boundary scan capabilities according to the JTAG standard. Once the microcontroller had received the input (which is the identity of the calling work station) from the decoder, the robot will move from the store house to its destination carrying the materials requested by that work station. This information is assumed to be communicated earlier between the work station and the store house. The movement sequence (the manner in which the robot has to respond after receiving the input) is structured as a program in embedded C and was burned into the microcontroller. The software used for creating the program was AVR STUDIO [12]. This software was provided by the ATMEL corporation.

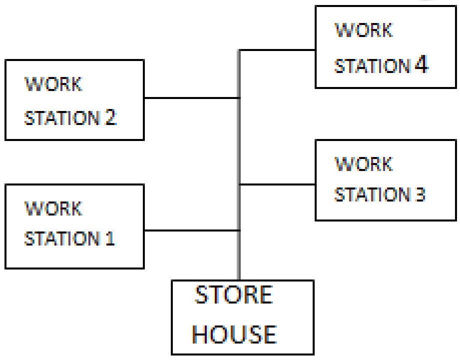
The microcontroller generates the output which control the gear motors. This output control the orientation in which the motors has to rotate. But this signal does not have the required voltage to drive the gear motor as the motor requires 12V for its operation and the peak voltage of the microcontroller output is 5V. Hence an interface is required between the two. This purpose is met using the motor driver

IC L293D [13]. It converts the 5V input (motor control bit) from the microcontroller to a 12V signal without altering the Phase and frequency of the input. This newly generated signal will drive the motors of the robot.

The robot moves from the source to its destination by following a dark line on a white background. This track should be kept clear of any obstacles. For sensing the black line, an array of 5 IR sensors [14] were fitted on the bottom part of the robot. The output from these sensors is given as input to the microcontroller alongside the second input from the decoder of the receiver. These two inputs will be used by the microcontroller to generate the control signal for driving the gear motor.

2. Output Evaluation

The robot was tested on a track with 4 work stations, whose layout is given below:



Every work station is provided with a switch for calling the robot. Each of these switches are configured as the input to the microcontroller. When a caller is pressing a switch, that particular input pin of the microcontroller is connected to the ground (binary 0) throughout the switch pressing duration. The remaining switches which were left unpressed will continue to provide a high voltage (binary 1) to the input pin. Here we assume that only one caller is pressing the switch at a particular instant of time. That is provision for accommodating multiple calls are not incorporated in this basic model.

The all possible input sequences and its implications are listed in the table below:

Table 1: Input sequence description

Input Sequence	Implication
1 1 1 0	Switch S1 at workstation 1 is pressed
1 1 0 1	Switch S2 at workstation 2 is pressed
1 0 1 1	Switch S3 at workstation 3 is pressed
0 1 1 1	Switch S4 at workstation 4 is pressed

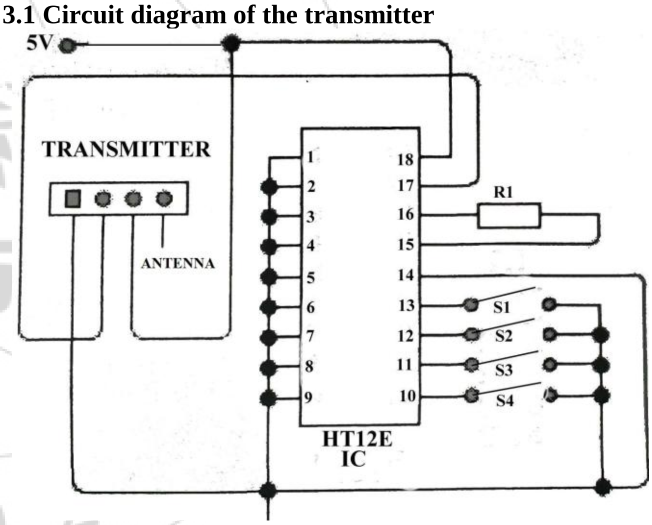
Initially the robot was tested by giving a valid input (listed in table 1) and trouble shooting was carried out. Each of the unexpected behavior shown by the robot was researched and studied in detail and was rectified by editing and modifying the software program burned into the microcontroller. The software code was refined consequently, until the robot behaved in the expected manner in all respects.

Firstly, the switch S1 in work station 1 was pressed keeping all other switches intact. Hence the input 1110 was transmitted through the transmitter. Upon its reception, the robot started to move from its default position (which is at the store house) to its destination, which in this case is work station 1. The robot after reaching the work station remained there for a predefined time and then turned back and returned to its default position.

Similarly the switches at each of the work stations were pressed one at a time and the robot was tested. In all the cases the robot moved from its default position to its destination, remained there for a specific time and returned to its default position which is the expected and pre defined result that the robot had to exhibit.

3. Circuitry

3.1 Circuit diagram of the transmitter



3.2 Circuit diagram of the robot

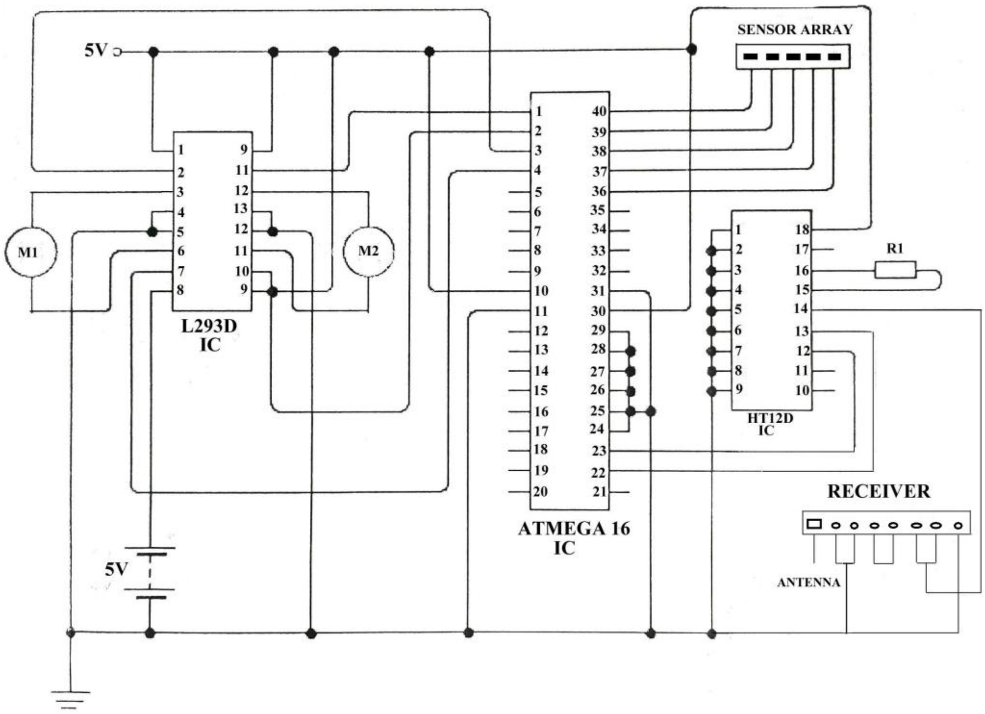


Figure 2: M1 and M2 denotes gear motors, R1 denotes I Kg ohm resistor

4. Embedded C Code Used In The Robot

The following software code was designed and structured to meet the expected predefined way of working, and was burned into the microcontroller for the desired functioning of the robot.

```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
DDRA=0x00;
DDRB=0xff;
DDRC=0x00;
int a;
int b;
int q,n;
while(1)
{
a = PINA;
q = ~PINC;
if(q= = 0x01)
{ n=1;}
if(q= = 0x02)
{ n=2; }
If(q= = 0x04// q= = 0x08//q= = 0x00)
{ n= 3;}
switch(n)
{
 case 1:
{
 a= PINA;
while(a!=0x07//a!=0x0f)
{
 if(a= = 0x04)
{ PORTB=0x05;}
if(a= = 0x08//a= = 0x10//a= =0x0c//a= = 0x18)
{PORTB=0x01;}
if(a= =0x01//a= =0x02//a= =0x06//a= =0x03)
{PORTB=0x04;}
a=PINA;
}
if(a= =0x07//a= =0x0f)
{PORTB=0x04;}
a= PINA;
while(a!=0x00)
{
if(a= =0x04)
{PORTB=0x05;}
if(a= =0x08//a= =0x10//a= =0x0c//a= =0x18)
{PORTB=0x01;}
if(a= =0x01//a= =0x02//a= =0x06//a= =0x03)
{PORTB=0x04;}
a=PINA;}
if(a= =0x00)
{PORTB=0x04;}
while(a!=0x1f)
{if(a= =0x04)
{POTRB=0x05;}
if(a= =0x08//a= =0x10//a= =0x0c//a= =0x03)

{PORTB=0x04;}
a=PINA;
}
if(a= =0x1f)
{PORTB=0x01;}
while(a!=0x00)
{
if(a= =0x04)
{ PORTB=0x05;}
if(a= =0x08//a= = 0x10//a= =0x0c//a= =0x18)
{PORTB=0x01;}
if(a= =0x01//a= =0x02//a= =0x06//a= =0x03)
{PORTB=0x04;}
a=PINA;
}
```

```
if(a= =0x00)
{ PORTB=0x04;}
_delay_ms(1000);
PORTB=0x00;
break;
}
case2:
{a=PINA;
while(a!=0x1c//a!=0x1e)
{if(a= =0x0x04)
{PORTB=0x05;}
if(a= =0x08//a= =0x10//a= =0x0c//a= =0x18)
{PORTB=0x01;}
if(a= = 0x01//a= =0x02//a= =0x06//a= =0x03)
{PORTB=0x04;}
a=PINA;}
if(a= =0x1c//a= =0x1e)
{PORTB=0x01;}
while(a!=0x00)
{if(a= =0x04)
{PORTB=0x05;}
if(a= =0x08//a= =0x10//a= =0x0c//a= =0x18)
{PORTB=0x01;}
if(a= =0x01//a= =0x02//a= =0x06//a= =0x03)
{PORTB=0x04;}
a=PINA;}
if(a= =0x00)
{PORTB=0x04;}
while(a!=0x1f)
{if(a= =0x04)
{PORTB=0x05;}
if(a= =0x08//a= =0x10//a= =0x0c//a= =0x18)
{PORTB=0x01;}
if(a= =0x01//a= =0x02//a= =0x06//a= =0x03)
{PORTB=0x04;}
while(a!=0x00)
{if(a= =0x04)
{PORTB=0x05;}
if(a= =0x08//a= =0x10//a= =0x0c//a= =0x18)
{PORTB=0x01;}a=PINA;}
if(a= =0x1f)
{PORTB=0x04;}
if(a= =0x01//a= =0x02//a= =0x06//a= =0x03)

{PORTB=0x04;}
a=PINA;}
if(a= =0x00)
{PORTB=0x04;}
_delay_ms(1000);
PORTB=0x00;
break;
}
case3:
{break;}
}}}
```

5. Future Enhancements

The robot designed was a prototype and its performance and efficiency can be improved significantly by adding several other additional features some of which are listed below:

1) Switch operated response at destination:
Currently the robot goes to its destination and waits there for a pre defined time and then automatically starts its return journey. This is the designated window for the working staffs to load/unload the materials to/from the robot. But practically this might be inconvenient, since the staffs may need more time to perform there loading/unloading operations. This problem can be solved by providing switch on the robot which the staffs at the destination can operate after completing there desired operations.

2) Pressure Sensor
The robot currently gives no indication whether it is loaded with more load than what it can bear. This issue can be addressed by providing a pressure sensor and alarm feature. As soon as the robot is overloaded (when the load pressure exceeds the predefined threshold pressure), an alarm is enabled indicating the current load is more than the capacity of the robot.

3) Priority Assessment:
Currently, if the robot moves to its destination then until it comes back to its default position, a second caller cannot place the request. This problem can be addressed by assigning priority to the callers. Suppose while the robot is moving due to the call initiated by a caller and while is engaged in this movement, if a second caller places a request then the second caller's identity can be stored in a stack and after the robot has finished the first caller's response, then from the stack it can automatically recover the identity of the second caller and can carry out that request. This feature can be incorporated by modifying the embedded C program in the microcontroller accordingly.

References

[1] Andrea Goldsmith, Wireless Communications [2004].
[2] Communication systems, 4th edition, Simon Haykin.
[3] ASK Superhet Receiver Module, ST-RX04-ASK receiver, Summit Technology Co Ltd, Available from the web address: < http://www.avrfreaks.net/sites/default/files/st-rx-04-ask.pdf> [September 2012]
[4] ASK Transmitter Module, ST-TX01-ASK receiver, Summit Technology Co Ltd, Available from the web address: <http://www.datasheetspdf.com/PDF/ST-TX01-ASK/665195/1> [September 2012]
[5] dBm, Available from the web page : <https://en.wikipedia.org/wiki/DBm> [September 2012].
[6] Chapter 4 of radio receivers, Superhetrodyne receiver, Available from the web page: <//www.mikroe.com/ old/books/book/ chapter4/ >[September 2012].
[7] HT12A/HT12E, 2^{12} Series Encoders, Holtek. [February 2009].
[8] HT12D/HT12F, 2^{12} Series Decoders, Holtek. [November 2002].
[9] CMOS VLSI Design: A Circuits and Systems Perspective, Revised 3rd edition, Neil H E Weste.
[10] 2466T–AVR–07/10 (Atmel 8 bit AVR microcontroller with 16 kilobytes in system programmable flash), Atmel Inc.
[11] Reduced instruction set computing (RISC) , Available from the web address :<https://en.wikipedia.org/wiki/Reduced_instruction_set_c

omputing> . [September 2012].

[12] Atmel AVR studio 5, Available from the web address:<//www.atmel.com/microsite /avr_studio_5 > .[September 2012]

[13] L293 , L293D quadruple half-h drivers SLRS008C –September 1986 – revised November 2004 .

[14] How to build an IR sensor, Available from the web address:<http://maxembedded.com/2013/08/how-to-build-an-ir-sensor/>. [September 2012].

Author Profile

Nithin Pradeep received his B.Tech. degree in Electronics and Communication Engineering from the University of Kerala in 2013.He is currently working as a Design Engineer in the Global design centre of Atkins (India) pvt ltd.