W.E.S.L.E.Y.

Waste Eliminating System for Lazy Engineering Youths

Special Sensor Report
December 9, 2003
John Mercado

The Special Sensor for my robot is the infrared beacon/RF system. The system consists of the following:

On the robot:

- A Laipac TLP-434 RF Transmitter
- A Holtek HT-12E encoder

On each beacon:

- A Laipac RLP-434 RF Receiver
- A Holtek HT-12D decoder
- A low Rds MOSFET
- 555 timer circuit

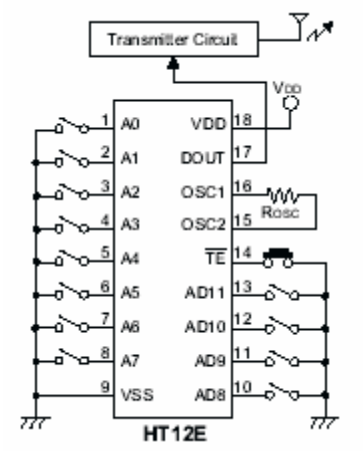

Figure 1

On the robot, the Holtek encoder is connected to a port of the ATmega128, I used port A. A similar circuit to the one I used to connect the HT-12E is shown in Figure 1. For the address pins on the chip, I connected A0 through A4 to ground, and A5 A6 and A7 to Port A pins 4, 5, and 6. For this system, I only used 2 different beacons, so the 8 bits of address information is unnecessary. I could have used just one address pin, but I used 3 for extra reliability. I connected the /TE pin to Port A, pin 7, although this could have been grounded to make it always transmitting. AD8 through AD11 I connected to Port A, pins 0 through 3. I could have also used one pin here, but I wasn't doing anything else with Port A, so I used all four data pins. The DOUT pin was connected to the RF transmitter. For Rosc I used a 2 MΩ resistor because it had to be slow. The maximum speed of the transmitter is 2400 bits per second.

On each beacon I had a Holtek decoder hooked up to the RF receiver. The circuit in Figure 2 is similar to the one I used. In order to create an address for each beacon, I chose either 5v or ground for pins A5-A7. For the home beacon I chose 011 and for the destination beacon I chose 010. The rest of the address pins need to be grounded. For Rosc I used a 150 kΩ resistor to make it fast because the oscillator frequency of the decoder is desired to be 50 times the frequency of the encoder. The graphs to choose a Rosc, along with all the other information you would need can be found in the datasheet for the chip. I only took the data from pin D8 to the rest of the circuit, D9-D11 are left unconnected.

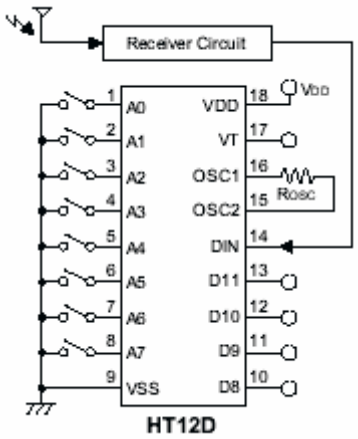

Figure 2

The signal on pin D8 is either high (beacon on), or low (beacon off). This is connected to the gate of a low Rds MOSFET. It is extremely important that the MOSFET have a very low Rds, otherwise you will lose a lot of voltage through the MOSFET and it's not a good switch. The drain of the MOSFET is connected to 5v, and the source is connected to the power bus for the 555 timer

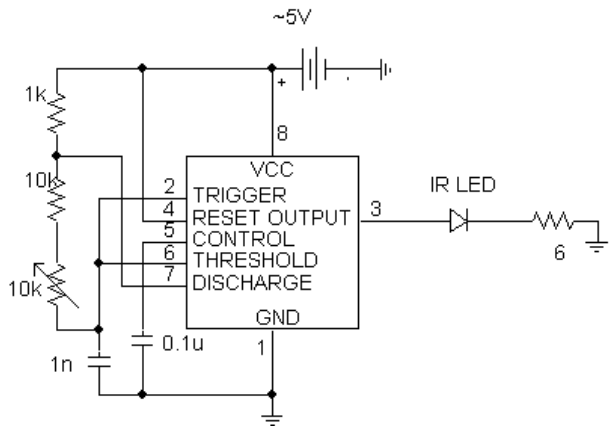

Figure 3

circuit. When D8 is high, current will be allowed to flow through the drain to the source. The result is a voltage controlled switch.

The 555 timer circuit is the same circuit found on the IMDL website. The only difference is I used a potentiometer as a variable resistor to fine tune the frequency of the square wave. The schematic for the timer circuit is shown in Figure 3.

The software for this system was simple, I just write to port A when I want to switch a beacon on or off. The address for the home beacon is 0b00000011 and the address for the destination beacon is 0b00000010. The connections to Port A are summarized in the table below:

Pin 7	Pin 6	Pin 5	Pin 4	Pin 3	Pin 2	Pin 1	Pin 0
/TE	A7	A6	A5	AD11	AD10	AD9	AD8

To turn on the home beacon all I need to do was write 0b0011xxx1 where the x's are "don't cares." To turn off the home beacon I write 0b0011xxx0. To turn on the destination beacon, I write 0b0010xxx1 to port A, and to turn it off, I write 0b0010xxx0 to port A. It's that simple.

This is a very effective way to use many IR beacons on the same frequency. If you used 11 bits of address information and 1 bit of data, you could have up to 4,096 different beacons on the same frequency.