University of Florida
Department of Electrical and Computer Engineering
EEL 5666 – Spring 2004

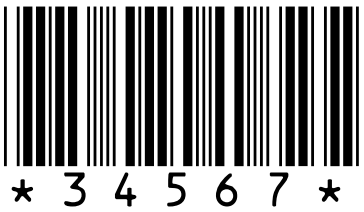
SPECIAL SENSOR REPORT: KANESCAN BARCODE SCANNER

Tuesday, April 20, 2004

Steven Pickles

THEORY

The scanner encounters and reads a valid barcode such as the following sequence:



This follows the Code 39 barcode scanning format, as it is an easily obtained and freely available font and also is easily read by the barcode scanner. Other formats may be used, but the presented application assumes the use of the Code 39 format. The asterisks at the beginning and end of the numbers 34567 indicate to the scanner the start and stop points of the data. Upon a successful read by the scanner, the numbers 3, 4, 5, 6, and 7 will be serially transmitted to the microprocessor.

OBJECTIVE

In this particular application, the barcodes will be encoded so that the robot will be able to determine its current location. Basically, the barcodes will act as road signs attached to the floor of a warehouse and will be read by the robot as it follows a line to its destination. By following the encoding scheme, it is possible for the robot to perform in a dynamic environment provided the barcodes are consistent. As a result, many different variables such as speed and next turn direction can be found “on-the-fly” by the mobile robot.

REFERENCES

Vendor Name: Kanecal
Part Number: Kanescan

Website:	www.kanecal.com
Phone Number:	877-KANECAL
Price:	$79.00 (they offer a student price of $59.00)

INTEGRATION

The barcode scanner can communicate with a microprocessor via USB, RS-232, or PS/2. While USB and PS/2 provide an easy way to interface the scanner for PCs, most embedded systems and microcontrollers generally have some form of serial port with which integration is the easiest. Out of the box, the scanner comes with a 66″ modified serial cable (in the case that a USB or PS/2 model is chosen, the cable type will vary) along with a 12″ PS/2 cable that provides power to the unit. At the opposite end of the cable is a 10-pin RJ-45 male plug. Despite having ten pins to work with, the proprietary serial cable makes use of only six. On the surface this may not seem to make sense, but remember that there are two other interfacing options for the scanner. The remaining four pins are most likely used with either USB or PS/2, and the universal connector allows for identical barcode scanners to be used in any one of the three possible interfaces. For setup instructions to program the scanner to transmit serially via RS-232, see appendix C.

Since the scanners are intended for use with point-of-sale (POS) computer terminals, the shipped cables are too long and too bulky for use with a small standalone robot. It is, however, possible to modify the cable or to create a new cable altogether. If the cable is modified, refer to figure 1, table 1, and the warning on the next page to be sure the barcode scanner will nto be damaged by the modification.

The current case that is presented consists of communicating with a microprocessor at 9600 baud, eight data bits, no parity bits, one stop bit, and no flow control. Therefore, only four wires are required to be used to receive a signal from the barcode scanner -- 5V power (VCC), ground (GND), transmit (TXD), and receive (RXD). The flow control signals, ready-to-send (RTS) and clear-to-send (CTS), may be ignored.

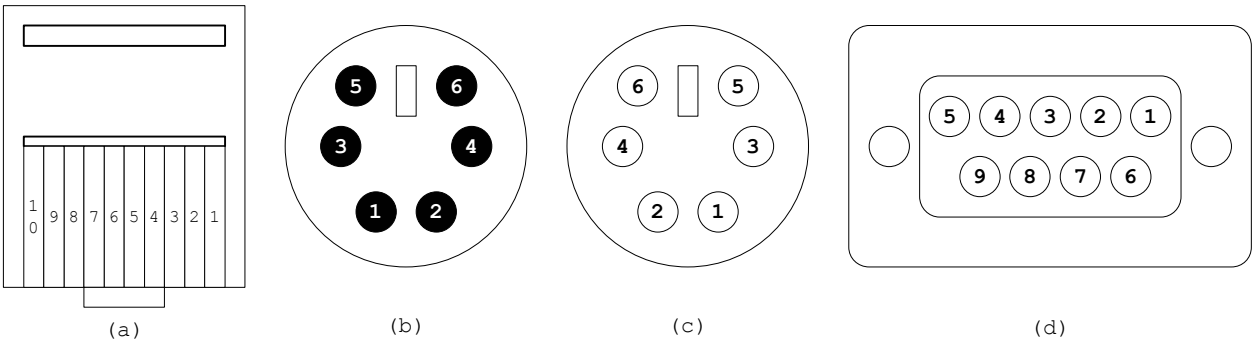


Figure 1 – Pin Placement for Included Cables

As can be seen in figure 1, there are a total of four connectors that are associated with the cables included in the box. Figure 1 (a) shows the 10-pin RJ-45 connector, (b) and (c) represent the 6-pin PS/2 male and female connectors, respectively, and (d) is the DB9 serial connector. Following each of the pinouts exactly is essential, especially if a replacement cable will be fabricated. **DO NOT** modify or create a new cable unless you connect the intermediate transmission signals through an RS-232 driver chip before connecting to the microprocessor. Failure to do so may render the barcode scanner inoperable.

Signal	10-pin RJ-45	6-pin PS/2 Male and Female	DB9 Female
GND	1	3	1, 5
VCC	2	4	9
No Connect	3		
No Connect	4		
No Connect	5		
No Connect	6		
CTS	7		7
TXD	8		2
RTS	9		8
RXD	10		3
GND Shield		3	5

Table 1 – Connector Pinout Arrangement

The original cable has an RS-232 driver embedded in a rectangular-shaped casing near the middle of the cable. Thus if the original cable remains fully intact, no RS-232 driver is needed. Refer to table 1 above to see which signals are connected to which pins of each connector.

Once the scanner has been successfully connected to the microcontroller, the scanner is ready to send data. The most logical way to handle incoming data from the scanner is with a receive interrupt because there is no guarantee that data will ever be received (no barcodes are scanned, for example). Therefore polling the receive flags is a waste of resources (unless implemented as part of a multitasking environment). By using a processor interrupt, it is possible to only worry about the data when it is incoming from the barcode scanner.

APPENDIX A - KANESCAN FEATURES

- light weight at just over 4 ounces
- wide 3” reading window
- can read barcodes up to 1.6” (40mm) away
- 55 to 100 scans per second
- smallest readable bar width of 0.004” (4 mil)
- optional LED and audible read confirmation
- 660nm wavelength red LED array
- 2048 pixel CCD array

APPENDIX B – SOURCE CODE

When a UART receive interrupt is generated, it is necessary to perform a function to handle the incoming data. Using an Atmel ATmega128, this is done with the following interrupt vector code snippet:

```
//     UART0 Receive Interrupt Vector
SIGNAL(SIG_UART0_RECV)
{
       //  Scans for a barcode, assuming the barcode has a length of 5
```

```
        if(scanCount < 5)
        {
                fromScanner[scanCount] = UDR0;

                //      Map the *CORRECT* ASCII value
                switch(fromScanner[scanCount])
                {
                        case(ZERO):
                                fromScanner[scanCount] = '0';
                                break;
                        case(ONE):
                                fromScanner[scanCount] = '1';
                                break;
                        case(TWO):
                                fromScanner[scanCount] = '2';
                                break;
                        case(THREE):
                                fromScanner[scanCount] = '3';
                                break;
                        case(FOUR):
                                fromScanner[scanCount] = '4';
                                break;
                        case(FIVE):
                                fromScanner[scanCount] = '5';
                                break;
                        case(SIX):
                                fromScanner[scanCount] = '6';
                                break;
                        case(SEVEN):
                                fromScanner[scanCount] = '7';
                                break;
                        case(EIGHT):
                                fromScanner[scanCount] = '8';
                                break;
                        case(NINE):
                                fromScanner[scanCount] = '9';
                                break;
                }
                scanCount++;
        }
}
```

APPENDIX C – RS-232 SETUP INSTRUCTIONS

In order to set the barcode scanner to transmit via RS-232, refer to the programming manual that is shipped from the manufacturer along with the scanner. In addition, follow the steps described here:

1. Set the default configuration:
   a. Scan the “Default” barcode on the top of page 9
2. Interfacing options:
   a. Scan the “Start” barcode on the top of page 10
   b. Scan the “RS-232” barcode in the middle of page 10
   c. Scan the “End” barcode on the bottom of page 10
3. End of text message:

a. Scan the “Start” barcode on the top of page 32
b. Scan the “None” barcode on the top of page 32
c. Scan the “End” barcode on the top of page 32

By default, after each successful scan the reader beeps. After a while this becomes annoying, but can be adjusted by using the barcodes on page 37. In addition, the scanning control can also be adjusted if there is a need to keep the scanner on all the time. Refer to pages 15 and 16 to achieve this functionality.

Before interfacing the scanner with the microcontroller, it is advisable to connect the scanner to a PC to ensure that the unit is functioning as expected. To scan directly into a terminal program, attach the device to the serial and PS/2 ports on the PC and then adjust the computer communications options to match those of the scanner. Scan a few test codes and if there is a problem try reprogramming the scanner. If all else fails, contact the manufacturer. They are helpful and available at care@kanecal.com, or via phone.