Magnetic Stripe Card Reader Project

Darcy L. Watkins - August 25, 2003

Introduction

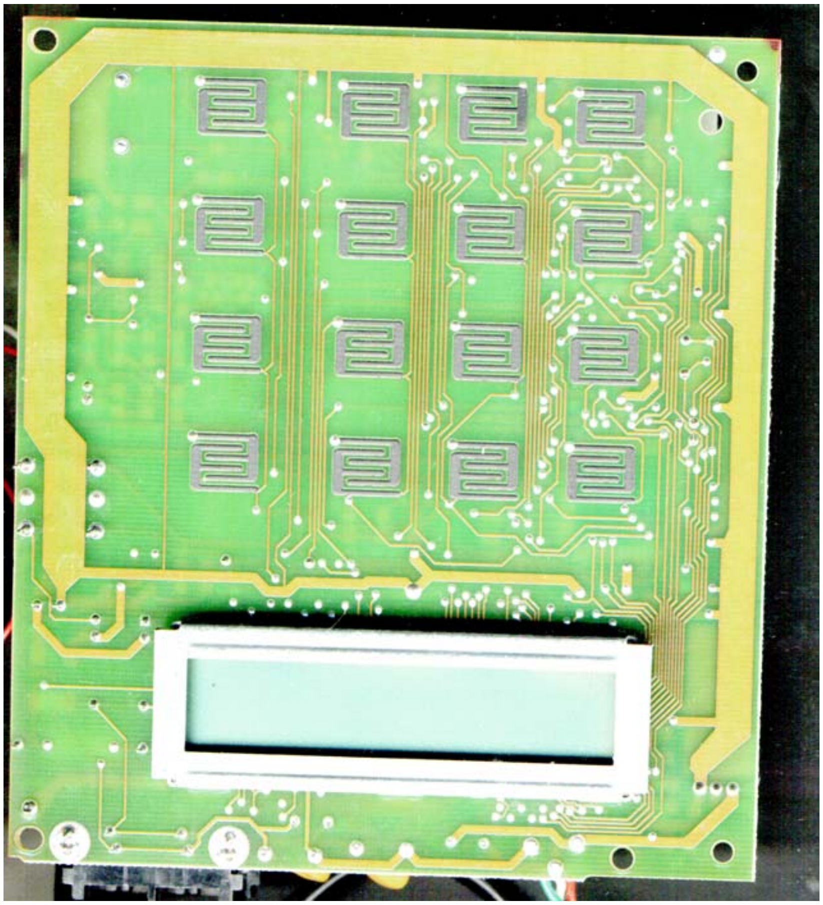
Discover (credit card company) issued a number of credit card validation terminals to merchants a while back, namely: Microbilt TAT 150. In its current configuration, these are not very useful, but they appear at surplus outlets at a very inexpensive price.

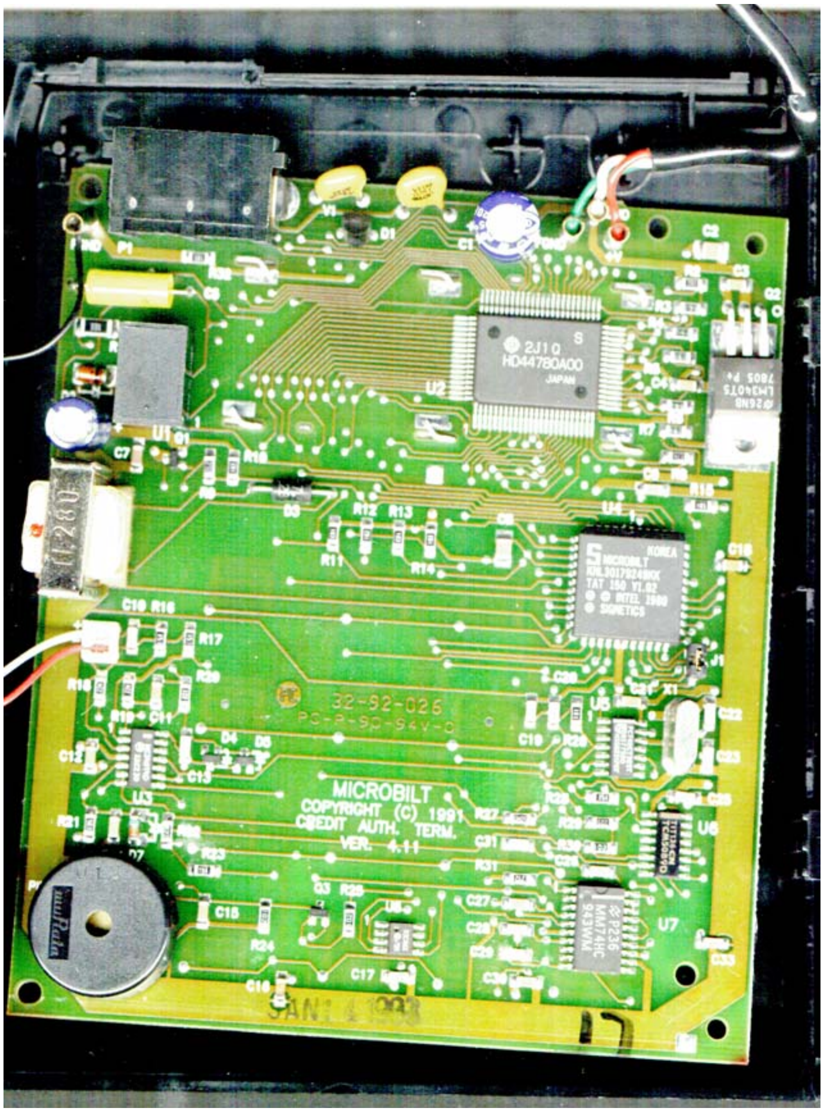
These terminals have a single track head, interface logic, an LCD controller, a 300 baud telco modem, 16 key keypad, DTMP encoder, a relay, buzzer, a serial EEPROM and a pre-programmed 80C51 family microcontroller.

The goal of this project is to replace the microcontroller with an AVR pin compatible device and to make a viable development platform which could be useful for card lock access and similar applications.

Pictures of the PC Board

Here are front and back pictures of the PC board from a scanner.





AVR MCU

At this time, the ATmega8515 appears to be the MCU of choice featuring 8k flash, 512 bytes of EEPROM (so that the external serial EEPROM isn't even necessary) and 512 bytes of SRAM. This MCU supports ISP (but not JTAG).

Programmability

The ISP port can be used with a minimal side effect that some buzzer noise may occur during programming. Wires need to be soldered to MCU pins for MOSI, MISO, SCK, RESET_ along with Gnd and Vcc to a 6 pin connector suitable for an ISP connection. Fortunately there are convenient places to solder on the wires.

Reset Circuitry

Necessary changes to reset circuitry is simple. Capacitor C8 needs to be replaced with a 4.8 k ohm resistor. C8 is well labeled and easy to spot.

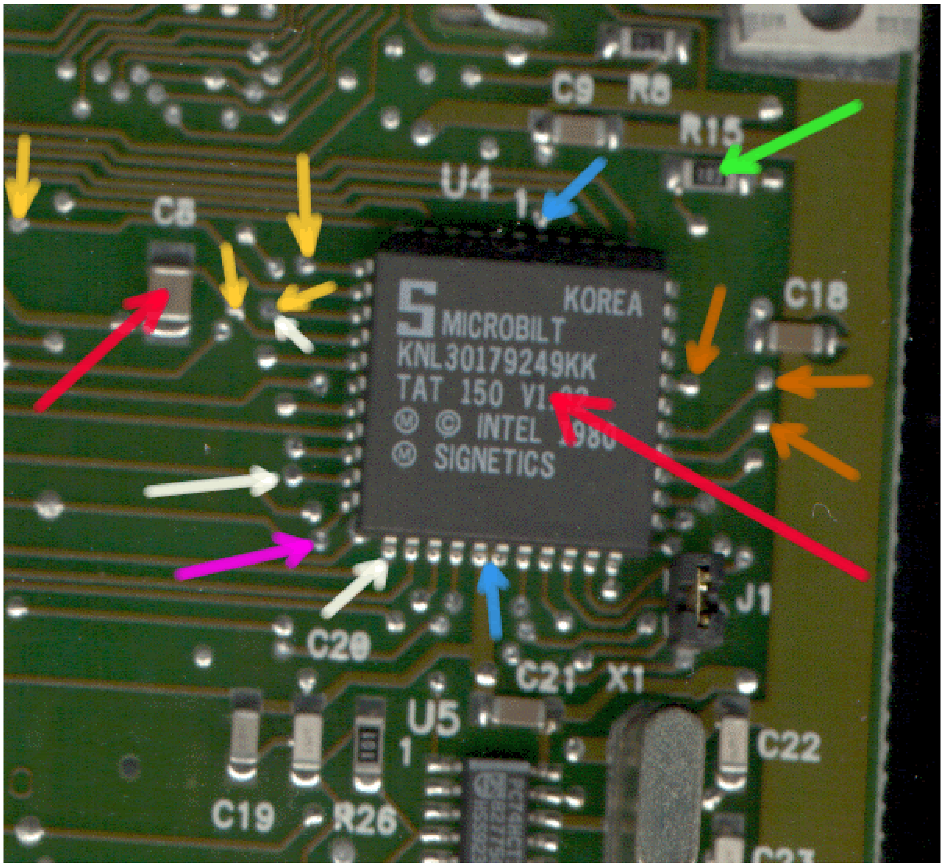
CPU Pin-Out Map (ATmega8515 - PLCC)

Pin	8051 Port	Connected to	ATmega8515 Port
1	N.C.	---	N.C.
2	P1.0/T2	Modem Squelch Transmitter (SQT)	PB0(OC0/T0)
3	P1.1/T2EX	Relay U1 via Q1	PB1(T1)
4	P1.2	LCD E	PB2(AIN0)
5	P1.3	LCD R/W_	PB3(AIN1)
6	P1.4	LCD RS	PB4(SS_)
7	P1.5	DTMF ToneDisable_	PB5(MOSI)
8	P1.6	Piezo Buzzer via Q3	PB6(MISO)
9	P1.7	EEPROM MWire Data Out	PB7(SCK)
10	RST	[to reset circuitry]	RESET_
11	P3.0/RxD	Modem Rx-Data	PD0(RxD)
12	N.C.	---	Vcc
13	P3.1/TxD	Modem Tx-Data	PD1(TxD)
14	P3.2/INT0_	Mag Head Signal In [U3-8]	PD2(INT0)
15	P3.3/INT1_	Modem Carried Detect (CD_)	PD3(INT1)
16	P3.4/T0	EEPROM MWire Data In	PD4(XCK)
17	P3.5/T1	Ripple Count 2-Q1 [U5-10 via R28]	PD5(OC1A)
18	P3.6/WR_	EEPROM MWire Serial Clock	PD6(WR_)
19	P3.7/RD_	EEPROM CS	PD7(RD_)
20	XTAL2	X1	XTAL2
21	XTAL1	X1 via J1	XTAL1
22	Vss	Gnd	Gnd
23	N.C.	[to Gnd]	N.C.
24	P2.0/A8	LCD DB0 / DTMF Row-1	PC0(A8)
25	P2.1/A9	LCD DB1 / DTMF Row-2	PC1(A9)
26	P2.2/A10	LCD DB2 / DTMF Row-3	PC2(A10)
27	P2.3/A11	LCD DB3 / DTMF Row-4	PC3(A11)
28	P2.4/A12	LCD DB4 / DTMF Col-1	PC4(A12)
29	P2.5/A13	LCD DB5 / DTMF Col-2	PC5(A13)
30	P2.6/A14	LCD DB6 / DTMF Col-3	PC6(A14)
31	P2.7/A15	LCD DB7 / DTMF Col-4	PC7(A15)
32	PSEN_	---	PE2(OC1B)
33	ALE	---	PE1(ALE)
34	N.C.	---	N.C.
35	EA_/Vpp	10k pull-up [R15] – can be removed	PE0(ICP/INT2)
36	P0.7/AD7	Keypad Col 4	PA7(AD7)
37	P0.6/AD6	Keypad Col 3	PA6(AD6)
38	P0.5/AD5	Keypad Col 2	PA5(AD5)
39	P0.4/AD4	Keypad Col 1	PA4(AD4)
40	P0.3/AD3	Keypad Row 4	PA3(AD3)
41	P0.2/AD2	Keypad Row 3	PA2(AD2)
42	P0.1/AD1	Keypad Row 2	PA1(AD1)
43	P0.0/AD0	Keypad Row 1	PA0(AD0)
44	Vcc	Vcc	Vcc

Hardware Modification Details

First of all, capacitor C8 and the microprocessor, U4, (shown by red arrows) need to be removed and replaced. Replace C8 with a 4.8 k ohm resistor. Replace U4 with an ATmega8515 MCU.

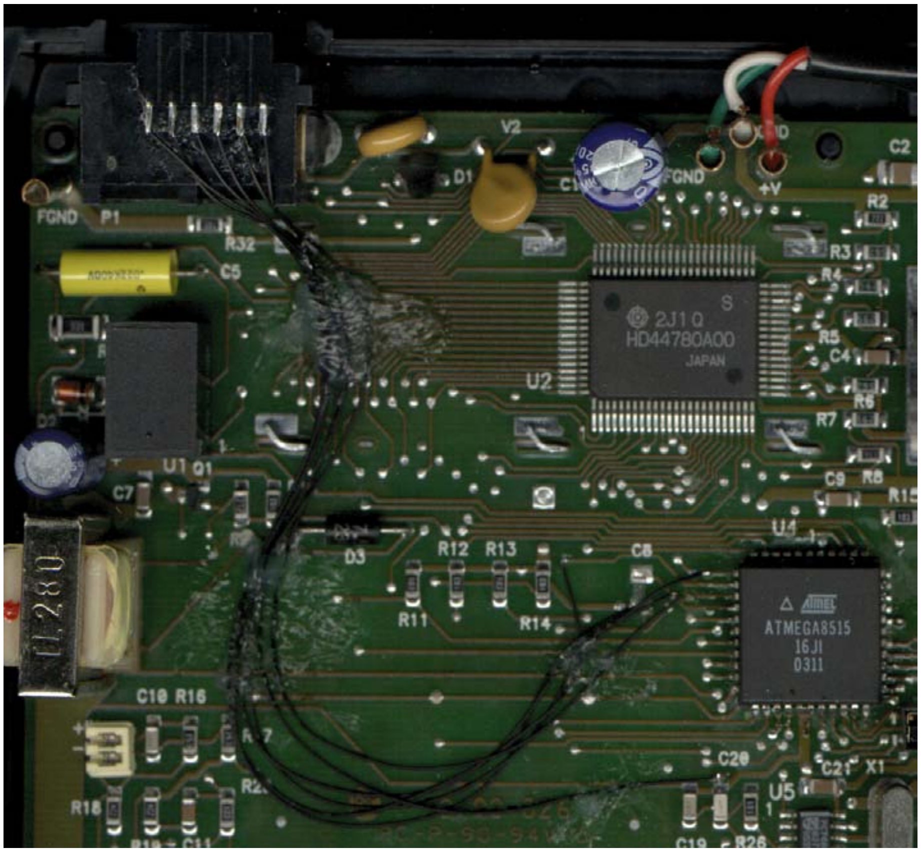
Next for an ISP interface, you need access to pins 7, 8, 9 and 10 of U4 (shown by yellow arrows) and you need ground and Vcc (shown at MCU by blue arrows but there are probably more convenient points to access these from).



The ATmega8515 is different from its 80C51 counterpart which allows a few expansion options open. MCU pins 32, 33 and 35 (shown by orange arrows) are available as spare I/O pins. Pin 35 has a 10k pull-up resistor, R15, (shown by green arrow) which may need to be removed depending on what you use pin 35 for.

Since the serial EEPROM device uses a microwire interface, you could connect other microwire devices using the shared I/O lines (and address up to three using the spare I/O pins mentioned earlier). The microwire lines are shown by the white arrows. The microwire data out is also used for the ISP interface as serial clock, but the chip select lines going inactive on reset should make this not a problem.

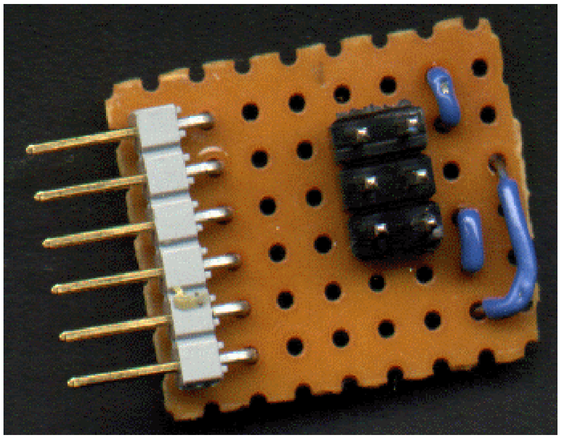
After Modifications



Above are the author’s modifications. Bottom right is the new MCU (and harder to see, the reset resistor which I had to use 5k2 instead of 4k8 which should work ok). I used a six socket single line header which I glued on top of the telco connectors to minimize the amount of mods needed to the plastic housing.

ISP Header Adapter

I made an adapter from 6 pin ISP to the six pin header as shown below:



The adapter is based on prototype board with horizontal copper traces. I cut the three rows through the six pin header and used the blue wires to route the back side. This results in the following pin-out from top to bottom.

1. Vcc
2. MISO
3. SCK
4. RST
5. Gnd
6. MOSI

This is for use with the Atmel AVRISP. Of course, if you are using a different device, you can change the pin-out accordingly.