EEL 4914C Final Technical Report Fall 2001

Dead Kelvin

By: Nicholas Ivano

Customers: Dr. Antonio Arroyo
Jason Grzywna

December 7, 2001 F4

Table of Contents

1. Project Description / Market

1.1 Project Scope:

The main focus is to develop a wireless-kill switch for an electric powered car. A printed circuit board will interface with existing hardware and will be used to detect and generate signals for electric motors and steering servos. The circuit will also be able to stop the car autonomously when it is required.

1.2 Project Summary:

We need to develop a printed circuit board (PCB) which can detect and generate PWM signals to control an electric cart called Kelvin. The most important function is to kill the motors to Kelvin either by a member of the engineering team or by default when there is a loss of signal to control the vehicle. The PCB must also have the ability to receive and transmit digital signals to and from the 68HC11.

The PCB which is also referred to as 'Dead Kelvin' will have an RS-232 9-pin interface, which will permit program code to be downloaded to the 68HC11. The interface will also allow communication between the client's terminal via the server's motherboard. An LED mounted on the PCB will display data being transmitted to the HC11 for debugging purposes. A MAX232 chip will be able to convert the RS-232 signal to a TTL signal.

The I/O interface consists of an 8255 PPI chip. The 8255 can be programmed to make various pins on the chip either inputs or outputs. We will want to be able to kill Kelvin whenever one of the inputs is active high. We would also like to take one the outputs high to drive an LED showing the status of the motors. The outputs must also be used to drive signals to the 8254 timer chips.

The 8254 timer can be programmed for many operations. We can use two of the modes to generate PWM signals that are required. One of the counters can generate the period of the remaining counters. Sending data to each counter's control register controls the duty cycle of the remaining counters. An additional digital output signal from the 8255 and the PWM from the 8254 will not only control the speed of the motors but the direction of each motor as well.

1.3 Target market (potential customers):

Anyone who requires a motor and servo control unit.
Anyone who needs I/O ports for a 68HC11 running in expanded mode.
Small scale robotics.
Anyone who requires a low-level control interface.

2. Customer Needs

2.1

Customer Needs Table	
Rank	**Customer Need**
5	Safely killing Kelvin's motors
4	Easy to interface
3	Small package (SOIC)
2	Free the 68HC11 resources for other tasks
1	Low power requirements

2.2 The customers existing platform is very difficult to interface. I recommended moving the functions of several boards on to one PCB with a simple RS-232 interface. Since the platform is mobile, I suggested using smaller surface mount chips, which require less power and would also lighten Kelvin's weight. All of this would have to be done without neglecting the most important task, which is to kill Kelvin as needed.

3. Project Specifications

Project Specification Table	
Project Metrics	**Value or Range**
Number of digital inputs for killing	8
Number of digital outputs for status	8
Number of PWM signals	5
Power input (unregulated)	12V
Serial port interface	DB-9 (RS232)
PWM signal period	21.25ms
Number of MIL style motor interfaces	4
Number of general purpose PWM signals	3

Brief description of each metric:

(1) The digital inputs could be used to kill Kelvin when any of the pins go high. The input port can also be used for general I/O purposes.

(2) The digital outputs will give a visual status to the design team on whether the car has been killed using a red LED. The output pins also controls various functions of the 8254 such as GATE control. The output port can also be used for general I/O purposes.

(3) Two dedicated PWM signals will control the motor drivers and a steering servo on the car. One of the PWM signals is connected to an optical isolator for motor control while the second PWM signal is connected to a steering servo. The steering servo only needs the PWM signal because the power and ground are supplied elsewhere.

(4) A 12-volt battery supply is regulated at a constant 5 volts for all on board chips. A 12-volt battery powers a DC-to-DC switching power supply, and then the switching power supply powers all electronic components on the car.

(5) The serial port allows communication between the 68HC11 and Kelvin's main computer. When code needs to be downloaded to the board, a serial cable connects Dead Kelvin's board with a PC. The cable is then connected to the server motherboard for wireless communication.

(6) The PWM, which is used to determine the state of the kill switch, is 21.25ms long. The 68HC11 is able to detect this pulse using the Input Capture system. When 3 periods are calculated, the 6811 determines if two of the periods are the same. If all periods have different elapsed times then the car's motors should be killed.

(7) The MIL style interface between the 68HC11 and the motor controllers and servos have 3 pins. One pin sends the PWM; one pin is the ground, while the last pin is 0 or 5 volt depending on the motor's direction.

(8) Three additional PWM signals exist for future expansion. It will be possible to add additional motors or servos at a later date.

4. Project Spec. Verification

4.1 Testing Methods Used To Verify Project Specs

(1) I connected the input port to a DIP-switch package and connected the output port to a series of LEDs. I wrote a simple program in ICC11, which outputs the input data (switch status) to the output port.

(2) I wrote a test program, which allowed me to enter a value for the duty cycle and channel. I connected a digital oscilloscope to the desired channel to detect a PWM. The O-scope can display the duty cycle, period, and frequency.

(3) The car's steering servo is damaged at this time so I found a scrap servo and controlled its position with my program. The motors were tested with the same PWM signal. I chose to use 14 different duty cycles to control the speed in one direction. Therefore, there are 28 total speeds to choose from.

(4) Dead Kelvin's prototype board has an interface for a 12V wall unit. All chip sockets were tested for 5V before the chips were placed onto the board. The connector from the switching power supply was checked for 12 volts before the board was powered up.

(5) The communications port was tested by shorting the Txd and Rxd pins on the TTL side of the max232 chip. When running in HyperTerminal, the keystrokes made should be displayed back onto the screen. I also used an LED on the board, which served as an indicator of the Dead Kelvin communicating with the outside world.

(6) I wrote a program, which printed the measured elapsed time. I noticed that the times were inaccurate, however they were consistent. Instead of comparing the time to what the elapsed time should be, my Input Capture ISR measures the elapsed time of the incoming signal three times. If two of the three times are close to the same, I allow the vehicle to continue. If all three are different then the car's motors are killed. I used a red LED to display when the car has been killed.

(7) I tested the headers for the motor direction by simply measuring the voltage of the pin, which controls direction. After I measured the correct voltage for the desired direction, I connected the motors directly to confirm the correct wheel direction.

(8) I tested the other PWM signals by using a program that allowed me to choose which channel I wanted change.

4.2 Extent to which each of the Project Specs was met

(1) The input port works well, however I did not write a routine in my code for detecting a high pin for killing the car. This is a simple routine that will poll the input port for the car's status. I'm not sure how often I need to poll the port until the car is fully operational; therefore I left it out for now.

(2) I know the output port works fine because the 8254 chips would not be able to operate properly without it. The motor direction control also works, which requires the output port to generate the proper signal. A low signal is forward and a high signal is reverse.

(3) All PWM channels are working as advertised. The board has two dedicated channels for the steering and the motors.

(4) The power for the board comes from a stable 12V switching power supply. The switching power source has 5V, 3.3V, 12V and –12V for various electrical devices. I could have used the 5V directly to power my board and done away with the power regulation on Dead Kelvin. Since one of the requirements was to make a general motor control board, I included the power regulation circuitry on my board so any future projects would not have to worry about generating clean source voltage.

(5) The communication from my board to rest of the world works well. I noticed a problem related to the clarity of signal to the IC1 pin of the 68HC11. As soon as I start to download code to the board, the signal from the radio receiver to the IC1 pin becomes very noisy. I partially solved this issue using a 74HC14 chip to clean the signal. If I later determine that the noise is coming from the PC, I may want to opto-isolate the signal for clarity.

(6) I generated the 21.25ms PWM by having a global variable in my code that I could change to get the proper period. I thought I may need to use trial and error to get the right period length, but the predicted counter value entered into the 8254 chip worked. I was

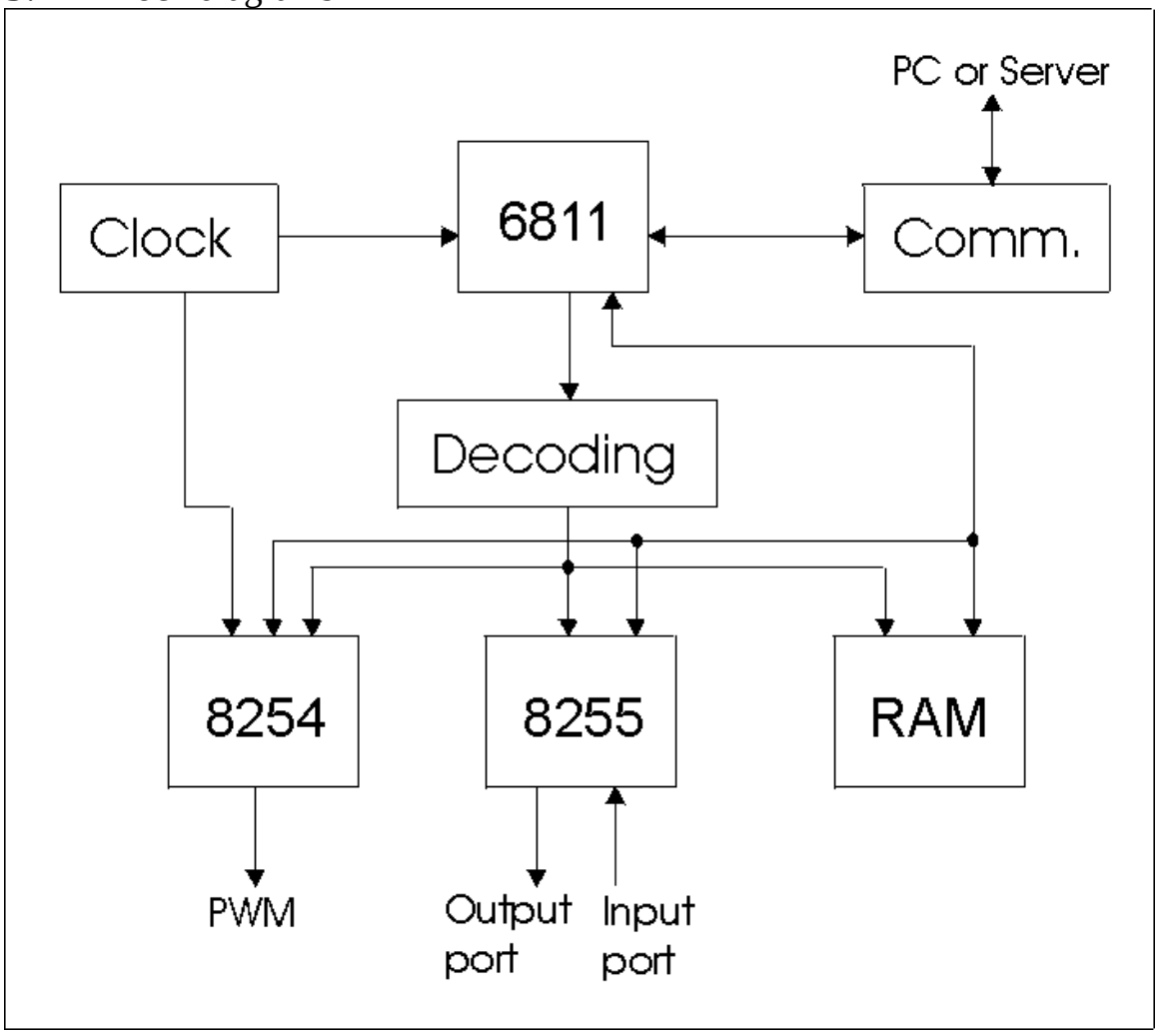
not convinced the input capture was measuring 21.25ms accurately and consistently. Since the issue of noise was raised in the previous metric I decided to take three elapsed time measurements and compare the similarity of those times. This turned out to be a more robust configuration.

(7) The MIL style headers for the dedicated PWM channel and the remaining three channels worked fine. I can connect the motors to any one of the headers and get the same speed and direction required.

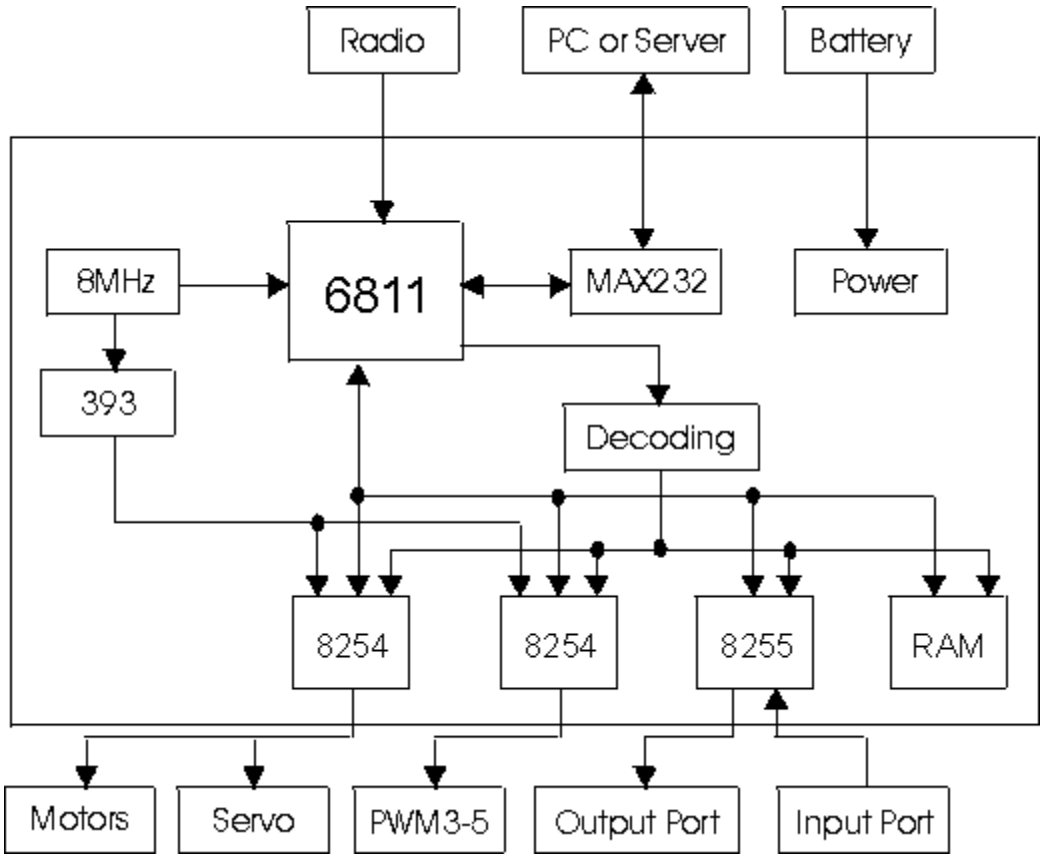
5. Project Details

Provide detailed information about various aspects of the Project such as:

5.1 Block diagrams



Preliminary Block Diagram



Final Block Diagram

5.2 Schematic – The final schematic in Protel is **Figure 1**.

5.3 Software – The code, which can be run in ICC11 independent of the server, is **Program 1**. The code, which allows the server to control the vehicle, is **Program 2**. The code in Program 2 does not allow the use of the kill switch at this time. Other issues in Linux have to be addressed before this is possible. Program 2 illustrates how the server will continually poll the Dead Kelvin board for its next instruction.

5.4 Package and assembly – Since Dead Kelvin will utilize surface mount chips for low power and small size, a professional PCB must be made for proper assembly. A special surface mount soldering station must also be used in order to ensure no shorts are created.

6. Project Cost

Part Description	Estimated Cost $	Actual Cost $
9-pin D-Sub connector	1.94	1.94
Reset Switch	0.35	0.35
oscillator (8MHz)	2.78	2.78

74138 decoder (SOIC)	0.52	0.52
PLCC socket	2.06	2.06
74138 decoder (DIP)	0.52	0.52
MAX232 converter	3.98	3.98
MAX232 converter (SOIC)	sample	4.10
74573 latch (SOIC)	sample	0.55
Double row male header	1.60	1.60
16 pin wire wrap socket	1.83	1.83
24 pin wire wrap socket	2.75	2.75
32 pin wire wrap socket	3.66	3.66
40 pin wire wrap socket	3.00	3.00
Wire wrap ID markers	2.00	2.00
0.1uF Capacitor (ceramic) qty-10	1.30	1.30
1.0uF Capacitor (electrolytic) qty-10	1.30	1.30
MC34064	sample	2.89
MC34164	sample	3.25
LED's	1.79	1.79
8254 timer	6.79	6.79
8255 PPI	6.50	6.50
7805 Voltage regulator	2.45	2.45
Various resistors	free	1.00
7408 (SOIC)	sample	0.38
7404 (SOIC)	sample	0.35
7400 (SOIC)	sample	0.35
100uF Capacitor (electrolytic) 16 volt	0.30	0.30
100uF Capacitor (electrolytic) 25 volt	0.39	0.39
power block	1.25	1.25
Motorola 68HC11E9	5.00	5.00
PC breadboard	2.50	2.50
Total Cost $	≈ $100.00	≈ $110

Since I developed a breadboard prototype first, the cost of the final product will differ substantially. The final product will not require any of the DIP chips or any of the wire-wrap sockets, however the cost of producing the printed circuit board professionally increases the cost an additional $75-$100. If more than two boards will be made the cost will decrease dramatically. The total projected price for making one board is ≈ $165.00.

7. Summary

Dead Kelvin is a low power, small package and expandable design, which generates and detects PWM signals. The board has an 8-bit input and output port available. A ripple counter with a four-position jumper can generate a 500kHz, 1MHz, 2MHz and 4MHz clock signal. Communication is handled through a 9-pin RS232 serial cable

and any 12V power supply can be used to power the board. Since most of all the 6811 systems are free to use, future expanse of the Kelvin project can be done with ease.

8. APPENDIX

Project Timeline (Gantt Chart)

	Wk1 9/16	Wk2 9/23	Wk3 9/30	Wk4 10/7	Wk5 10/14	Wk6 10/21	Wk7 10/28	Wk8 11/4	Wk9 11/11	Wk10 11/18	Wk11 11/25
Parts	X										
Comm.	X	X	X								
Brains		X	X	X	X						
Timers					X	X	X				
Input / Output					X	X	X				
Integration								X	X	X	X

Subproject 1 (Parts)
Acquire all parts necessary to build Dead Kelvin.

Subproject 2 (Communications)
Design and build a communications port, which allows the microprocessor to communicate with Kelvin's motherboard. This same port will also be used to download code to the 68HC11 processor. A max232 chip will be used to convert the RS232 level voltages to TTL and vice versa. An LED will be added in parallel with the data stream. This will help debug any download issues. The interface will be a DB9-pin connector.

Subproject 3 (Brains)
This refers to the installation of the Motorola 68HC11 microprocessor. The 6811 must be soldered in place and tested to ensure there are no shorts. Additional components must be added to the 6811 for proper operation. A reset switch with under-volt sensing capabilities is required. Regulated power and pull-up SIP resistors for the SCI system will be needed. Jumpers for mode operation and the SPI system are also needed. 128k X 8 memory will be added as well. The 6811 will have to detect when the signal is no longer available. This will be done using the Input Capture feature of the processor.

Subproject 4 (Timers)
The 8254 and 8255 will be required to generate the PWM signals. Software is a large part of this subproject. Software must be written in order to have the timers and PPI chips work in various modes. The 8255 is needed to control the gates in each timer.

Subproject 5 (Input / Output)
The 8255 needs to be programmed in order to get the proper amount of inputs and outputs to the 68HC11. Since the 6811 will be run in expanded mode, we will have to rely on the external I/O device. Software is a large part of this subproject.

Subproject 6 (Integration)
After all components are tested individually on the board, I must ensure they all work together without any problems. The last task will be to fully integrate the board into the car. Any remaining issues unresolved will be addressed at this point.

Important Additional Information:

Code is downloaded using ICC11. Before downloading any programs the chip must be in bootstrap mode (MODA & MODB are on) and the reset button pressed. After code has been downloaded the board must be put into run or 'expanded' mode (MODA & MODB are off) and the reset button is pressed again. If single chip mode is required (MODA is on & MODB is off).

Addressing info:

	PPI	Timer1	Timer2
Control Register	$6003 or $63ff	$6403 or $67ff	$6803 or $6bff
Input Port (A)	$6000 or $63fc	-	-
Output Port (B)	$6001 or $63fd	-	-
Output Port (C)	$6002 or $63fe	-	-
Counter0	-	$6400 or $67fc	$6800 or $6bfc
Counter1	-	$6401 or $67fd	$6801 or $6bfd
Counter2	-	$6402 or $67fe	$6802 or $6bfe