



**Two-Stage Model Rocket**  
**Joe Peck**  
**Brad Bruce**

## Overview

A two-stage model rocket consists of two sections – the booster and the sustainer. The booster section has the main motor for the rocket that supplies the initial energy for the flight, and the sustainer contains a motor that propels the rocket even higher. The flight of a two-stage model rocket consists of three main phases. In the first phase of flight, the rocket motor in the booster (1<sup>st</sup> stage) launches the entire rocket skyward. In the second phase, once the booster motor has burned out, the rocket motor in the sustainer (2<sup>nd</sup> stage) ignites which propels the rocket higher. The motor ignition also causes the booster to separate from the sustainer and tumble safely to the ground. In the third phase, when the sustainer has reached apogee, it deploys a parachute and the rocket floats gently to the ground. For this project, we constructed a two-stage model rocket that electronically controls the ignition of the rocket motor in the sustainer as well as the deployment of the parachute.

## Theory of Operation

The heart of the electronics consists of a small FPGA in which we instantiated a soft-core processor. The processor in the FPGA runs the control algorithm and the logic in the FPGA interfaces to the I/O. The program running on the soft-core processor is programmed in C and the FPGA logic is programmed in VHDL. To make the electronics more flexible for future projects, we designed two small PWBs. The main board contains the FPGA, configuration flash, oscillator, linear regulators, and additional support circuitry. The daughter board contains the accelerometer, low resistance FETs, bi-color LED, and additional support circuitry. Everything is powered from a single 9V NiCad battery. A simplified block diagram of the overall electronics is shown in Figure 1.

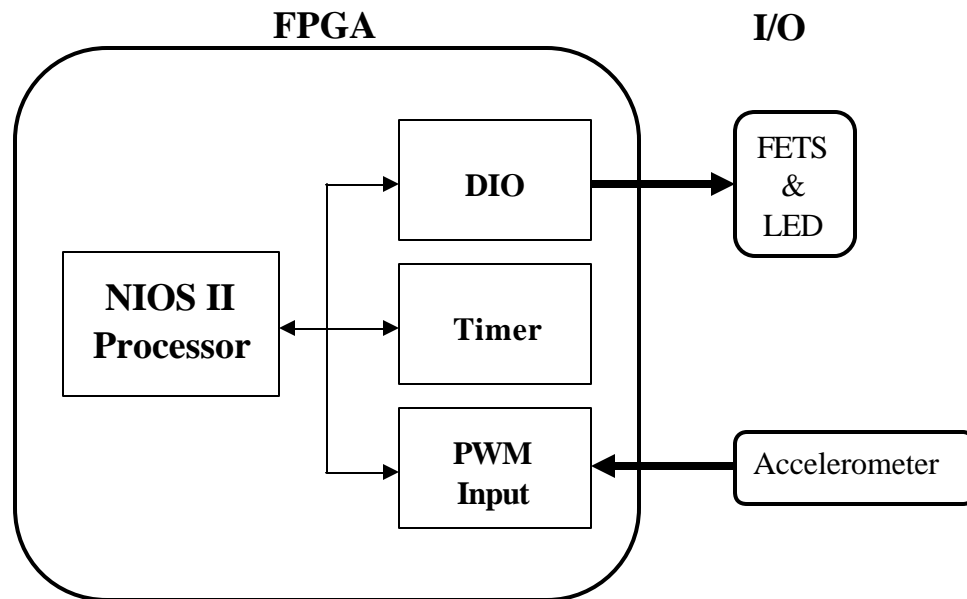


Figure 1: Electronics Block Diagram

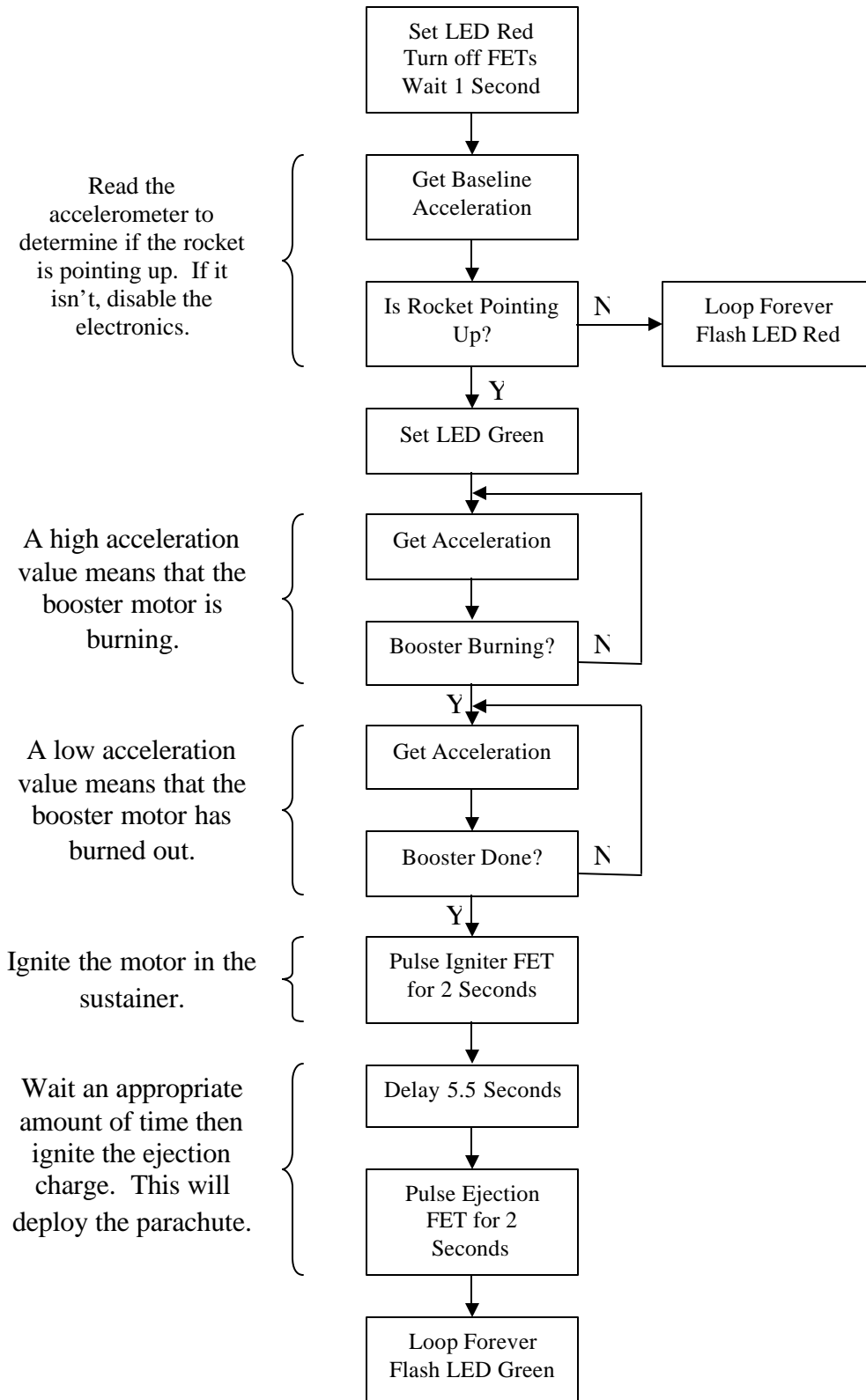
The rocket motor in the booster of the rocket is ignited from a ground-based launch controller. The rocket motor in the sustainer needs to be ignited after the booster motor burns out. We detect this event by using a MEMS accelerometer. The accelerometer generates a pulse width modulated (PWM) signal proportional to the acceleration of the rocket. We implemented a PWM input peripheral in the FPGA as a “custom peripheral” to the soft-core processor that allows the soft-core processor to read the current acceleration. In order to detect the burnout of the booster motor (to know when to ignite the sustainer motor), we first look for the acceleration of the rocket due to the booster motor burning. Once we see this high acceleration, we start looking for the acceleration to die down which corresponds to booster motor burnout. To eliminate false triggering due to shock (i.e. someone bumping the rocket while it is armed sitting on the launch pad), the processor filters the acceleration data using a smoothing filter (averaging filter). To implement a smoothing filter, we created a custom instruction for the processor that performs specialized fixed-point division in hardware for the final divide in each averaging operation (there is not a native instruction in the NIOS II Instruction Set Architecture for the particular type of specialized fixed point hardware divide we implemented).

Once the booster motor burns out, the electronics must ignite the rocket motor in the sustainer. The igniter requires several amps of current to ignite, so a low  $R_{DS(on)}$  FET is used to connect the 9V battery across the igniter. The igniter used for this particular rocket motor has a typical resistance of 0.8 ohms. The gate of the FET is driven by a 3.3V output from the FPGA so it is important for the FET to have a low  $R_{DS(on)}$  at a fairly low  $V_{GS}$ . A NiCad battery is used because it can provide high current for short periods of time due to its relatively low source impedance.

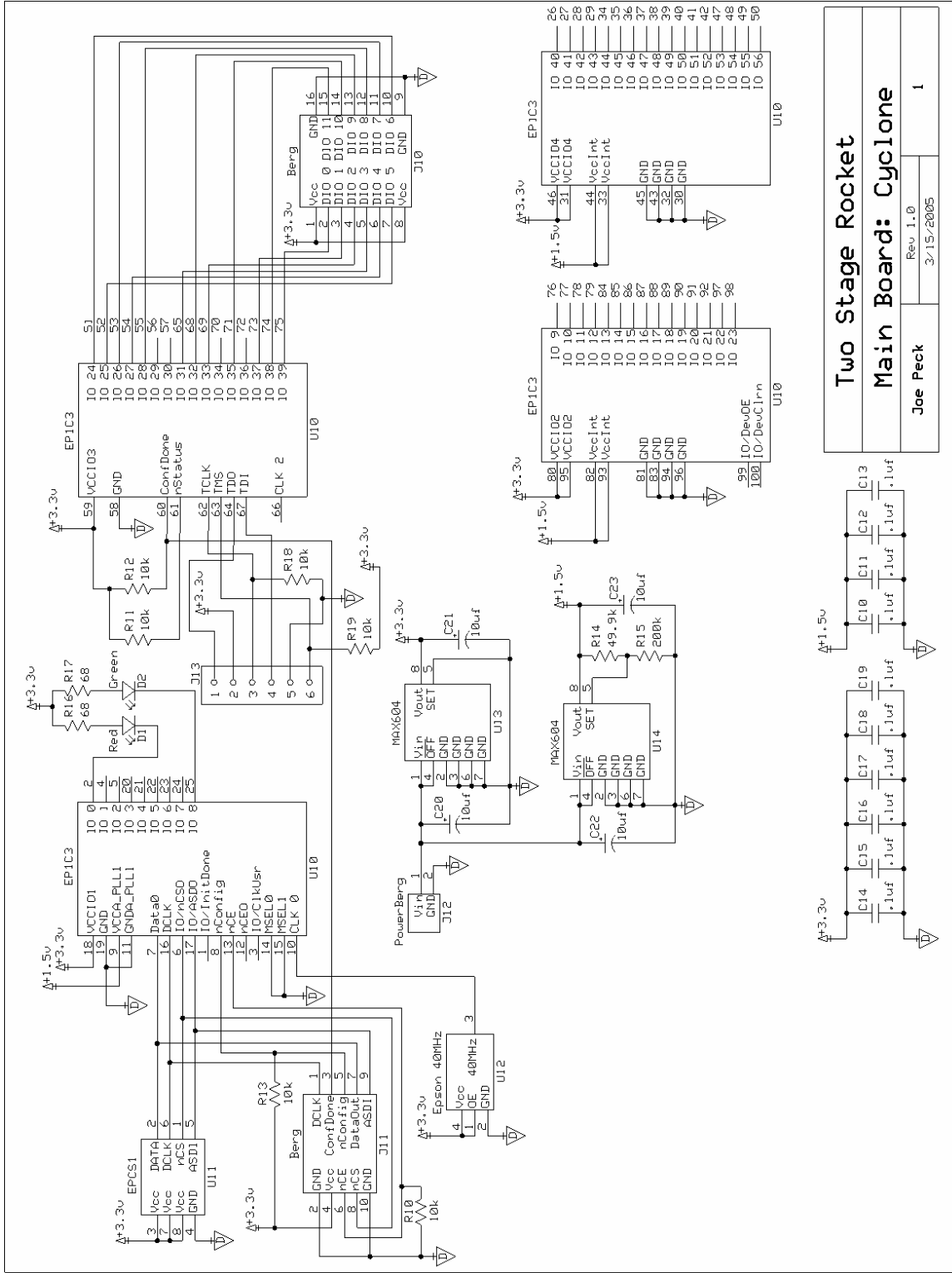
After igniting the sustainer, the electronics must wait to deploy the parachute at the appropriate time. We used a commercially available rocket simulation program to determine how long the rocket would take to reach apogee. We programmed the electronics to generate an appropriate delay before setting off the ejection charge. The ejection charge is a small amount of black powder that is ignited by an electric match, which is similar to the igniter described previously. The electric match is connected to a second FET controlled by the FPGA. The ejection charge forces the parachute out the nose cone end of the rocket, which will bring the rocket safely back to earth.

A bi-color LED is used to indicate various status conditions, such as ready to launch, electronics armed while the rocket is not pointing up, ejection charge fired, etc. The processor controls the LED and FETs from a digital output peripheral implemented in the FPGA logic. We also implemented a timer peripheral so that the processor does not have to rely on timed software loops to create the various delays required in the design.

## Software Algorithm



# Main Board Schematics



<b>Two Stage Rocket</b>	
<b>Main Board: Cyclone</b>	
Joe Peck	Rev. 1.0
	3/15/2005
	1





## Bill of Materials

### Main Board

Description	Allied #	Ref Des
10 k ohm, 0805	895-2676	R10 - R13, R18, R19
49.9k ohm, 0805	895-2440	R14
200k ohm, 0805	895-2406	R15
68 ohm, 0805	895-2892	R16, R17
0.1uF, 25V, ceramic, 0805	213-0110	C10 - C19
10uF, 16V, 3528, tantalum	644-1026	C20 - C23
Red LED, 0603	670-0002	D1
Green LED, 0603	670-6701	D2
2x8 pin header	518-2233 <sup>1</sup>	J10
2x5 pin header	518-2233 <sup>1</sup>	J11
1x3 pin header, middle pin pulled	518-2225 <sup>2</sup>	J12
6 pos mini header (debug only)	None	J13
Altera EP1C3 Cyclone FPGA	None	U10
Altera EPCS1 Configuration flash	None	U11
40MHz Osc, Epson SG8002CA	None	U12
Max 604, linear regulator	None	U13, U14

<sup>1</sup> 518-2233 is a 2x36 header. Cut it down to create the 2x8 and 2x5 headers.

<sup>2</sup> 518-2225 is a 1x36 header. Cut it down to create the 1x3 header.

### Daughter Board

Description	Allied #	Ref Des
0 ohm, 0805	895-2671	R20
121k ohm, 0805	895-2689	R21
330 ohm, 0805	895-2793	R22
0.1uF, 25V, ceramic, 0805	213-0110	C20 - C22
220uF, 10V, 7343, tantalum	213-1076	C23
Schottky diode	273-7523	D20
8 pos screw terminal	409-0038	J21
Bicolor LED	None	D21
16 pos berg receptacle	None	J20
3 pos berg receptacle, middle pin pulled	None	J22
ADXL210E	None	U20
IRLR7843	None	F20 - F22
Switch	None	SW20, SW21

Note that several of the components listed in the “None” category could be replaced with components offered by Allied with minimal redesign or verification. For example, the ADXL210E could be replaced by an ADXL202E with minor firmware changes.