FlexController™ Advanced System

4/1/5

Contents

Flextek Electronics
Carlsbad, CA 92008
www.flex-tek.com

1.0 FlexController™ Overview — Patent 6697685

Low-cost control and automation products that are easy to use:
Programming a microcontroller is time-consuming and tedious, but commanding FlexController™ SOC within the resource-rich environment of PC graphical development tools is fast and easy. Download the free FTview™ Active X software from www.flex-tek.com and install it on your Windows PC, then quickly generate custom applications in Visual Basic. Application notes with education kit and tutorial make it easy to quickly satisfy a broad range of requirements in power, thermal, motion, lighting, and flow.

Flextek Education Kit:
Kit provides intuitive feel to closed-loop control while demonstrating Visual Basic programming, SPICE simulation, microcontroller familiarization, and general electronic principles. Everything needed to perform valuable experiments, including FlexController™ SOC, FlexBus™ Communication board, IR Demonstration Board, FTedkit software, and FTview™ software.

FlexController™ System-On-Chip:
Single chip solution opens a window to the outside world from your PC. A free Active X Control extends the advantages of simplistic Visual Basic programming to external devices, combining the useful peripherals of a microcontroller with the convenience of a PC. Powerful mixed-signal hardware with intuitive software enables easy control, automation, and data acquisition in power, thermal, motion, lighting, and flow applications.

FlexBus™ Isolated Multi-Drop Interface:
Connects up to four FlexController™ SOCs to PC for communication through each COM port or USB to Serial Converter (Full Speed). The electrical isolation of the data bus eliminates line drops and ground noise from measurements, protects the PC from power transients, and enables operation at voltages not possible with other systems. Interface board with controller socket may also be used with PIC micro (ICD compatible).

MultiDriver™ Power Board:
The versatile power, control, and data interfaces of the MultiDriver™ Board save valuable time by providing an integrated system for digital controller development. Circuit board combines FlexController™ SOC and FlexBus™ Interface with a power stage that accepts 12V to 48V and provides 6A half-bridge or 3A full-bridge output current. Power board with controller socket may also be used with PIC micro (ICD compatible).

FTview™ Software:
Free software written in Visual Basic demonstrates capabilities such as data acquisition, plotting, and logging through a GUI interface for communication with FlexController™. An Active X Control is provided to write custom programs for interactive control and recording. Free download includes FTVdemo sample application, FTview™ Active X Control, and FTedkit for Education Kit.

FlexController™ Advantage

The electronics market has countless variations of data acquisition products, but these devices do no more than passive observation. Active PC-based control systems are highly desirable for automation but are often too expensive and time-consuming to learn. The essential elements for automation include an output power stage, isolated data interface, and custom programmability. Power is adjusted until the desired response is acquired over an interface that is isolated for safety and noise immunity. Since every system is different, it is critical that the controller be capable of custom configuration quickly and easily.

FlexController™ SOC combines the essential peripherals of a microcontroller (ADC, PWM, Timers, etc.) with the ease of Visual Basic (or C++) programming. Chips may be purchased separately for high-volume cost-sensitive applications or with convenient development boards to save valuable time. The FTview™ software may be downloaded free with demonstration applications that illustrate data logging and closed-loop control.

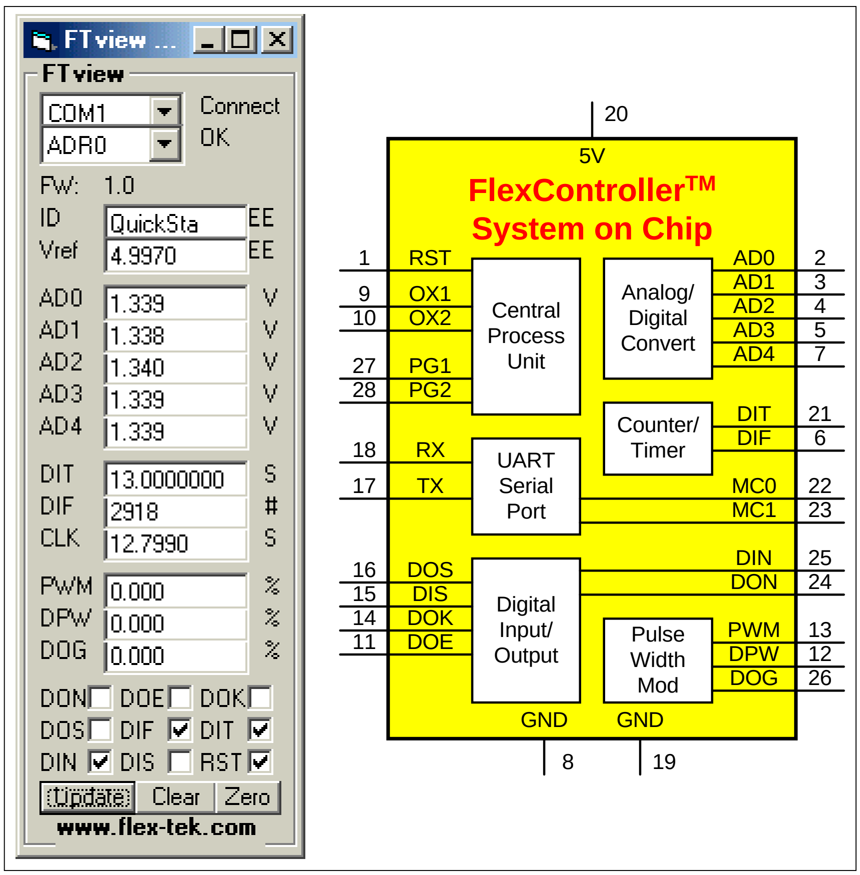

Figure 1.1. FTview™ Active X provides Intuitive Interface to FlexController™ SOC

Sample Visual Basic code for easy control:

```
Private Sub FTview1_NewData()                          ' Executes when data ready
    Thermal = 100 * FTview1.AD0volt                    ' Read AD0 voltage
    RPM = 60 / FTview1.DITinterval                     ' Read DIT timer
    FTview1.PWMduty = PIDout                           ' Write PWM duty cycle
    FTview1.UpdateCmd = True                           ' Update Control SOC
    Write #1, Time, Thermal, RPM, PIDout               ' Save data to File
End Sub
```

MultiDriver™ Board contains a versatile power stage that can be configured as a high power full-bridge or half-bridge driver. Control and communication is accomplished with FlexController™ System-On-Chip that measures sensor response and issues power command. PC data communication is safely handled through the isolated multi-drop interface FlexBus™. The graphical user interface may be quickly customized for a variety of applications with the FTview™ Active X. Together, these elements make affordable PC-based automation that is easy to use.

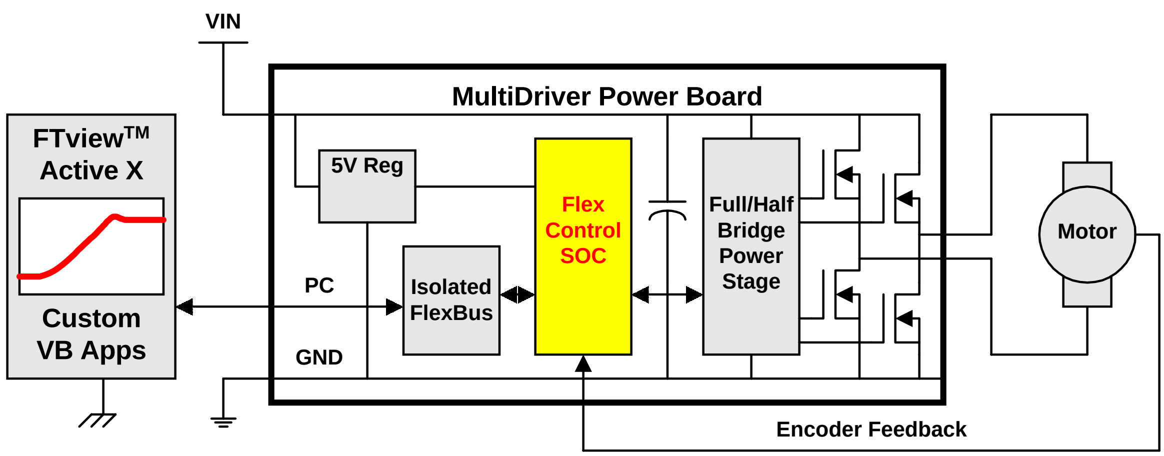

Figure 1.2. MultDriver™ Board with FlexController™ SOC in PC-Based Motor Control Example
(System may also be used for power, thermal, lighting, and flow control.)

Data communication to FlexController™ SOC is through FlexBus™ isolated multi-drop serial bus that eliminates line drops and ground noise from measurements, provides protection from power transients, and enables operation at voltages not possible with other systems. The data bus is port-powered, addressable, and compatible with common USB to Serial Converters for maximum flexibility.

The versatile power, control, and data interfaces of the MultiDriver™ Board save valuable time by providing an integrated system for digital controller development. Typical applications include power, thermal, motion, lighting, and flow control for industrial and educational purposes. Flextek product manual provides application notes with schematics and code examples.

Flextek Advanced Systems	Leading Competitor
Real-time control and graphical programming	Real-time control and graphical programming
Low-cost control chips and boards	Expensive laboratory equipment
Free software	Thousands of dollars for software
Tens of dollars for hardware	Thousands of dollars for hardware
Volume production and industrial automation	Laboratory experimentation
Few components to satisfy countless tasks	Countless components to satisfy few tasks
Easy to learn in an afternoon	Challenging to learn in a month

Flextek products have been featured in EDN, Electronic Design, Power Electronics Technology, and Nuts & Volts magazines in articles featuring innovative technology.

2.0 Flextek Education Kit Part FCED010

The Flextek Education Kit is a low-cost system that provides an intuitive feel to closed-loop control while demonstrating Visual Basic programming, SPICE simulation, PIC micro familiarization, and general electronic principles. Read entire manual for complete familiarization with products and technology used in kit, or just read next page and take it for a quick test run. Additional Education Kit details are in the Electronic Design Magazine article following this Quick Start Guide.

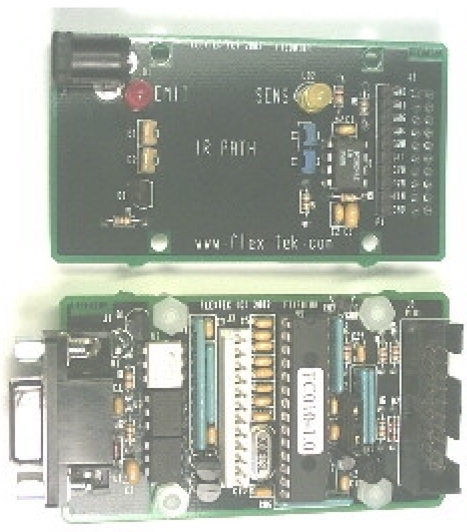

Figure 2.1. Education Kit Interface and Infrared Boards

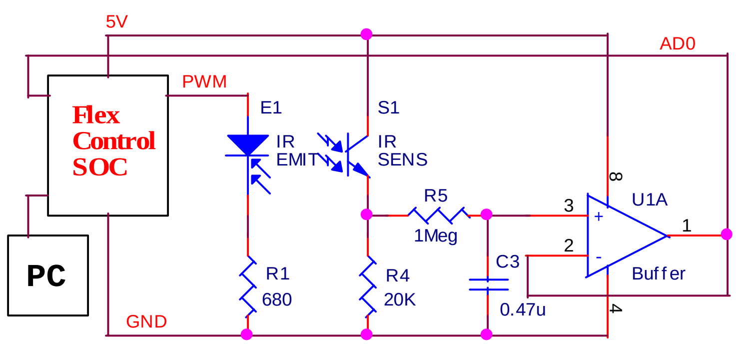

Figure 2.2. Education Kit Simplified Schematic

Flextek Electronics developed the digital controller in Figures 2.1&2 for educational purposes. The goal of a closed-loop controller is to maintain a desired response despite system changes and disturbances. Typical examples include heater temperature, motor speed, light intensity, and fluid flow. This kit uses a simple infrared circuit for stimulus-response behavior.

FlexController™ SOC drives current through an infrared emitter with its Pulse Width Modulator and measures the sensor voltage with its Analog to Digital Converter. The gain and time constant of the infrared circuit are set by resistors R4 and R5, respectively, which are selected for full ADC swing with timing in the range of human recognition. This circuit reacts to external light or shadowing and has visual feedback to provide an intuitive "touch and feel" for educational control experiments.

Quick Start Guide

Kit Contents:
1) FlexController™ System-On-Chip FCIC010
2) FlexBus™ Interface Board FCIF010
3) Infrared Demonstration Board
4) FTedkit Software download from www.flex-tek.com

Installation and Setup:
1) Purchase Flextek Education Kit at www.flex-tek.com and download FTview™ Software
2) Plug Infrared Board into top of FlexBus™ Board ensuring 20-pin connectors are properly aligned
3) Connect DB9M-F data cable from PC COM Port to FlexBus™ Board
4) Plug 9V or 12V power adapter from AC outlet to power jack of Infrared Board
5) Open FTedkit Software located in FTVdemo™ Folder of PC after running Setup

FTedkit Program: Software to demonstrate custom programming and closed-loop control. Infrared Board contains IR Emitter driven by FlexController™ PWM while the IR Sensor is measured by AD0. Click through the four program screens to get a quick feel for Education Kit operation.

Interface Screen: Example VB programming with FTview™ Control for FlexController™ SOC
a. Try FTview™ Control by typing PWM duty cycle, clicking Update, and reading AD0 response
b. Read and test code control examples in Software Interface section on right of screen
c. Click Auto Update with 50% PWM then Update and pass thin object through IR Path of Infrared Board
d. Turn on and off a incandescent light in the vicinity of the IR sensor
e. Notice that PWM drive remains constant while AD0 response varies in open loop mode

Control Screen: VB closed-loop code and infrared schematic for demonstration
a. Review code and schematic for basic understanding of closed-loop demonstration
b. VB code is written to automatically vary PWM driving AD0 response equal to desired setpoint

Demonstration Screen: Closed-loop demonstration of Infrared Board
a. See how easy it is to achieve desired AD0 of 2.500V by setpoint selection
b. Vary setpoint with scroll bar and observe automatic adjustment of PWM drive for desired AD0 response
c. Pass thin object through IR path and switch incandescent to see AD0 response held constant over time
d. Monitor PWM test point on scope to appreciate automatic duty cycle adjustment in closed-loop mode
e. Un-check Closed Loop for challenge in getting AD0=2.500V then pass object and switch light
f. Re-check Closed Loop and try different Proportional (KP) and Integral (KI) gains to observe responses

Simulation Screen: SPICE simulation schematic with equations for software control of infrared circuit
a. Download free evaluation version of SPICE from www.intusoft.com if no simulation software on PC
b. Simulate IR circuit with PI control and compare results to hardware response
c. Generate transient response and loop gain for various values of KP and KI

Caution: Ensure Infrared Board is properly aligned to 20-pin connector of FlexBus™ Board before applying power. If excessive voltage is inadvertently applied to 5VO of FlexBus™ Board, the 5.6V zener diode D2 (1N5232B) should protect the micro but may require replacement if over-stressed.

Additional Experimentation

Read the entire manual to appreciate the possibilities of Flextek products then try different experiments and applications. An interesting project is to use the PWM pin of the Education Kit to drive the Linear Power Amp shown in FlexController™ SOC section of manual. This amplifier could drive a small fan placed in the IR Path of the Infrared Board, breaking the IR beam with the fan blades to trip the DIT timer and DIF counter of the FlexController™ SOC. The infrared response is buffered by an op-amp and passed directly to DIT and DIF pins in Figure 2.3 for velocity measurement with PWM=100%. The motor speed code in FTview™ section could be modified to adjust PWM for constant fan speed as a motion control experiment. Pins AD1, DOG, and AD4 are also brought out as test points for the thermal or lighting experiments shown in FlexController™ SOC section.

Additional development exercises are possible with Flextek Education Kit. Modify the resistor and capacitor values in Infrared Board to vary gain and timing, then repeat original IR control experiment. Attempt to develop your own digital control algorithms in place of the PI example and compare results. The kit can even be used to test PIC micro programs in PIC16F876 or PIC18F2320 chips (same pinout as FlexController™ SOC). Create new applications based on your unique experience and design requirements then contact Flextek at www.flex-tek.com with your ideas to benefit others like yourself.

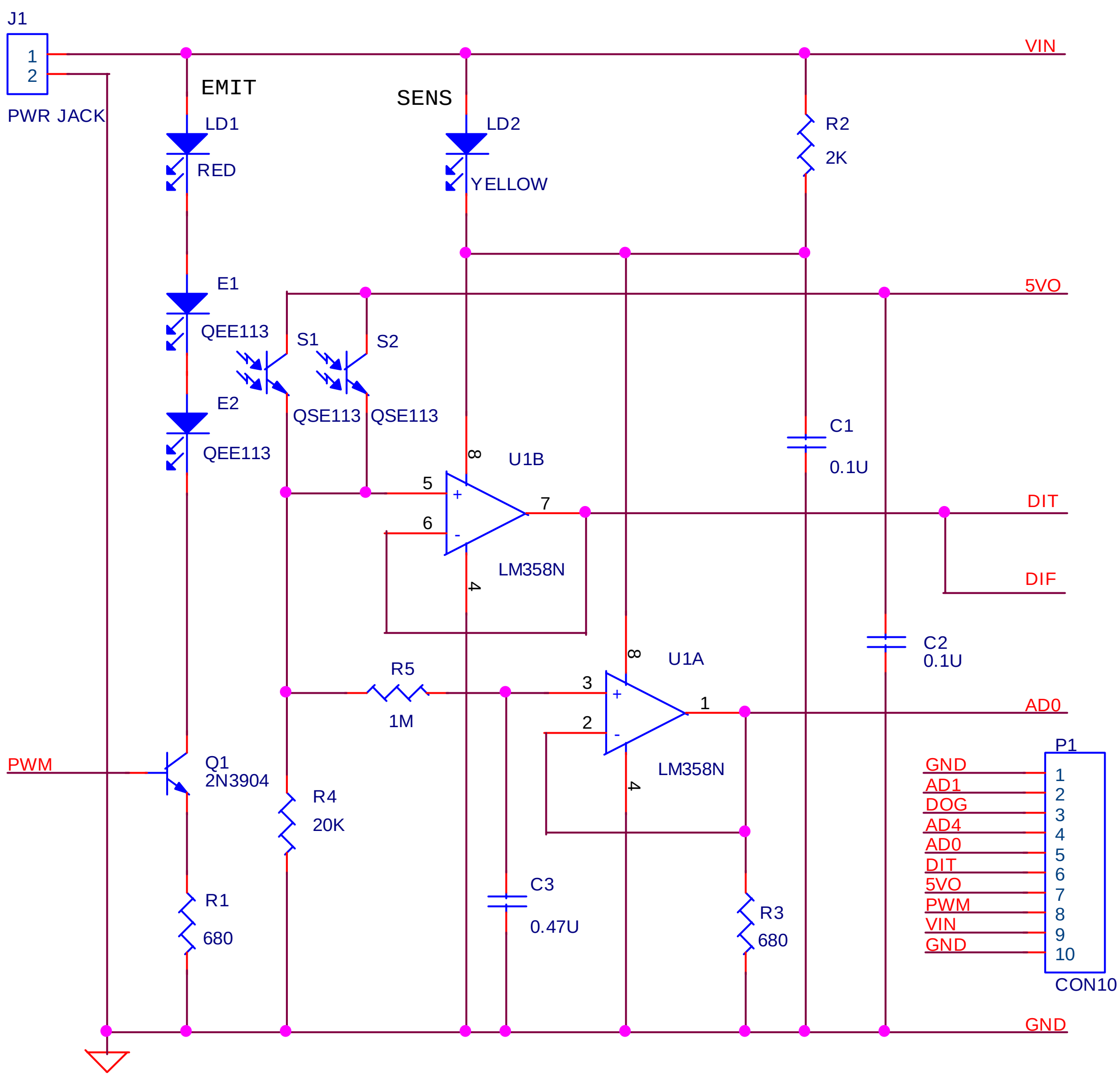

Figure 2.3. Infrared Controller Detailed Schematic

"Analog Simulation Tools Benefit Digital-Control-Circuit Designers"
Design Solution Article in Electronic Design Magazine December 4, 2003

Popular analog tools can be used to analyze common digital controllers to leverage decades of proven knowledge. In this example, SPICE is used to simulate a Proportional-Integral closed-loop controller written in Visual Basic.

Digital vs. Analog Controllers

Digital controllers have many benefits over their analog counterparts, including reduced parts count, greater flexibility, and ease of modification. However, something valuable has been cast aside in the transition from analog to digital, decades of useful knowledge. Countless articles have been written about analyzing analog circuits to optimize bandwidth and stability margin, while an equal number of articles have been written about tuning digital controllers through trial and error approaches that often result in less than optimal performance.

Although analog and digital controllers may appear vastly different, their principles of operation are usually quite similar. Therefore, popular analog tools like SPICE can still be used to benefit common digital PI (Proportional-Integral) controllers through analysis, without spending hours on complex math or a fortune on specialized software. Merging proven analog and digital technologies may achieve the best of both worlds.

Digital Controller Example

The goal of a closed-loop controller is to maintain a desired response despite system changes and disturbances. Typical examples include heater temperature, motor speed, light intensity, and fluid flow. Software driven microcontrollers with mixed-signal peripherals often accomplish these tasks. The PI software algorithm is popular for closed-loop control since it is a direct software adaptation of traditional op-amp circuits.

Flextek Electronics developed the digital controller in Figure 2.1-3 for educational purposes. FlexController™ SOC drives current through an infrared emitter with its Pulse Width Modulator and measures the sensor voltage with its Analog to Digital Converter.

The gain and time constant of the infrared circuit are set by resistors R4 and R5, respectively, which are selected for full ADC swing with timing in the range of human recognition. The PWM drive current illuminates the red LED (EMIT), while the supply current of the op-amp with burden resistor illuminates the yellow LED response (SENS). This circuit reacts to external light or shadowing and has visual feedback to provide an intuitive "touch and feel" for educational control experiments.

Figure 2.4 is the Visual Basic interface to the digital infrared controller that allows different PI gains to be evaluated for a fast stable response with minimal overshoot. This PC screen shot illustrates how easily the system may be adjusted for various applications, which is one of the primary advantages of a digital controller.

The benefit of a properly tuned control loop can be observed by noting the tall spike on the PWM drive signal with absence of spiking on the ADC response. The infrared emitter is initially driven hard to quicken the sensor response, but softens to steady state before the sensor can overshoot its desired value. Too much gain will cause the system to oscillate while too little gain results in a sluggish response. Two independent gains complicate coincident optimization and the third gain in PID make it even more challenging to properly tune. The Differential term is not required for most dominant-pole systems so it is omitted in this demonstration.

Gains are typically adjusted empirically rather than analytically because math in the S and Z domains tends to be cumbersome and mixed-signal control software is often expensive. A few simple approximations, however, enable this digital controller to be simulated with common analog software for systematic optimization of PI gains.

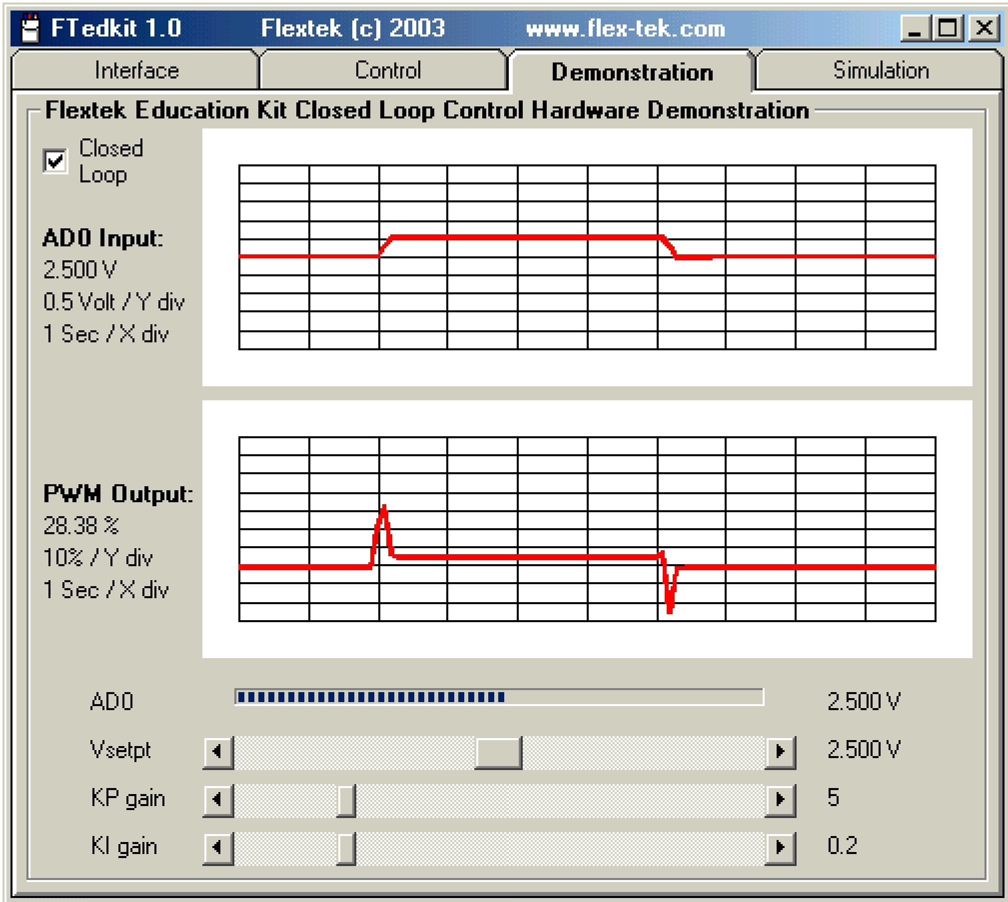

Figure 2.4. Visual Basic Interface Program to Digital Infrared Controller

Analog Simulation of Digital Controller

One of the most popular programs for simulating analog circuits is SPICE. Advantages of this program include decades of validated operation with countless application notes and ease of access including free student versions.

Early SPICE versions possessed limitations, including lack of digital simulation, so third-party vendors introduced upgraded versions with increased capabilities. Despite mixed-signal upgrades, there are still advantages to defining systems in terms of simple analog components, including portability between software packages and the intuitive understanding associated with universally defined parts.

A common and useful component in SPICE is the VCVS (Voltage-Controlled-Voltage-Source). It is essentially an ideal op-amp that amplifies and buffers a voltage signal with programmable gain. It is identified in schematics and netlists by a reference designator that begins with the letter "E". The VCVS in SPICE can be used to simulate the digital infrared controller of Figure 2.3 with the addition of resistors and capacitors. The utilization of these generic components enables the circuit to run in virtually any analog simulator by any vendor.

Figure 2.5 shows the analog equivalent of the digital infrared controller executing the PI algorithm. The upper portion of the schematic is the infrared hardware from PWM to AD0, and the lower portion is the PI control software from AD0 to PWM. Together, these circuits simulate a digital closed-loop controller in SPICE. Although this particular circuit was developed to simulate the digital infrared controller, its generic architecture may be used to simulate a wide range of analog and digital closed-loop control applications.

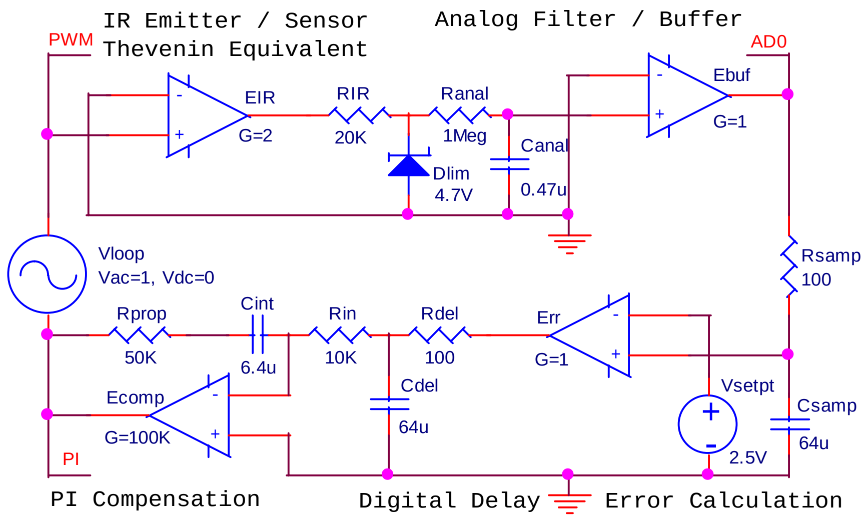

Figure 2.5. SPICE Analog Equivalent of Digital Infrared Controller

The precise gain and bandwidth of the infrared circuitry is not readily apparent from inspection of component data sheets, as is the case with many real systems. Operational performance is application dependent and subject to variations in manufacturing and environment. Therefore, the infrared gain and bandwidth is derived empirically from the open loop step response of AD0 to a sudden change in PWM.

The software interface program in Figure 2.4 allows the PWM drive to be changed instantaneously while observing the infrared response at AD0 in the actual hardware. A 10% duty cycle step change in the PWM drive yields a 20% (1V) change in the AD0 voltage response, which indicates an open loop gain of two. It takes approximately 0.47 seconds for the response to reach 63% ($1-e^{-1} \approx 0.63$) of its final value, which is expected since the dynamic response of the circuit is dominated by the RC filter (1MΩ • 0.47µF = 0.47Sec).

The infrared transistor in Figure 2.3 effectively acts as a current source driving a 20KΩ load from a 5V supply. The Thevenin equivalent of the emitter and sensor combination is then represented by the VCVS designated EIR with a gain of two and 20KΩ impedance limited to 4.7V by a zener diode. The RC filter is reproduced directly and the LM358 buffer is replicated with the unity gain VCVS designated Ebuf. The remaining circuitry is the PI closed-loop controller.

The PI algorithm is executed through the PC in Visual Basic for development ease since high level programming in a resource rich PC is faster and easier than generating firmware in a resource limited processor. The advantage of this approach is that a variety of control algorithms can be tested and compared quickly prior to committing to an embedded design. The penalty of this approach is slower updates, which is limited by serial data transfer time and PC interrupt latency. It is interesting to note that despite higher data rates, USB is usually slower in simple interactive applications due to extensive software overhead in a time-shared polled system.

The unique aspect of this software to analog conversion effort is to replicate operation of the PI software algorithm, which is not that difficult since PI is a direct adaptation of an op-amp circuit. However, one challenging aspect of this effort is to simulate sampled discrete operation of a digital system with continuous analog circuits, so reasonable approximations are employed for simplicity.

An average sample interval of 12.8mS is measured that includes reading micro inputs, calculating new control values, and writing micro outputs. This sample time is simulated with a 6.4mS RC network Rsamp/Csamp that is a reasonable approximation based on Nyquist criteria that defines a relationship of two between signal and sample rates. It is usually used to determine the required sample time for a given analog signal, but this application requires the inverse operation. The delay network Rdel/Cdel is added since the microcontroller contributes an additional sample cycle delay due to its ADC over-sampling and digital filtering for enhanced accuracy and resolution.

The unity gain error-amp Err takes the difference between the desired set point and measured response to be compensated for desired dynamics by an the op-amp Ecomp with proportional and integral gains. Ecomp has a high open loop gain of 100K so that its closed-loop gain is set by the ratio of feedback impedance Cint + Rprop to input impedance Rin. These amplifiers are configured such that a single inversion occurs throughout the control loop for negative feedback. Control textbooks usually use separate op-amps for each P and I gain term, but practical circuits combine them for reduced parts count.

The primary purpose of integral gain, which is a function of Cint and Rin, is to have high gain at low frequencies for low steady-state error. The capacitor in the feedback of the op-amp reduces the gain of the amplifier at higher frequencies to avoid instability. At these higher frequencies the impedance of Cint is much less that Rprop so the proportional gain set by the ratio of Rprop to Rin dominates over the integrator. This technique eliminates the phase lag contributed by the integrator that could otherwise induce instability when combined with the inherent phase lag of the infrared circuit.

To calculate the values of Cint and Rprop it is necessary to review the PI control software. It is written in Visual Basic as follows:

```
Vsetpt = 2.5: KP=5: KI = 0.2      ‘ Initialization
Verr = Vsetpt – AD0               ‘ Measured error
IntSum = KI * Verr + IntSum       ‘ Integrator running sum
PI = KP * Verr + IntSum           ‘ PI calculation
PWM = PI * 100/5                  ‘ Convert PWM (100%) from ADC (5V)
```

Loop every Tsamp = 12.8mS data sample update.

The SPICE analog conversion of this Visual Basic PI code is:

Rprop = KP * Rin = 5 * 10K = 50KΩ
Cint = Tsamp / (KI * Rin) = 12.8mS / (0.2 * 10K) = 6.4uF
Csamp = Cdel = Tsamp / (2 * Rsamp) = 12.8mS / (2 * 100) = 64uF

The SPICE circuit also contains two independent voltage sources. Vsetpt is varied in the time domain for transient response and Vloop is used to evaluate stability in the frequency domain. These sources serve as tools to quantify behavior of the circuit.

SPICE Results

Loop gain is the ratio of AC response (PI in Figure 2.5) to AC stimulus (PWM in Figure 2.5) through the feedback loop. A system is stable with less than 180° lag at unity gain or 0dB. Phase margin indicates the additional lag before instability is reached. The loop starts at 180° in a negative feedback system so 180° lag actually occurs at 360°, which is the same as 0°.

This stability criterion can be interpreted to state that the gain through the loop should roll off to unity in response to a single pole since each pole contributes 90° total phase lag and multiple uncompensated poles can induce stability. The infrared filter contributes one pole and the integrator capacitor contributes another, so the pole of the integrator is canceled by the proportional term prior to the unity loop gain frequency.

Figure 2.6 is the loop gain of the simulated infrared controller, which shows 3.25Hz bandwidth and 72.3°phase margin. These values were calculated automatically by the post-processor in the evaluation version of Intusoft SPICE. Mathematically the ratio of PI to PWM is taken as a difference when calculating gain in dB, as well as phase.

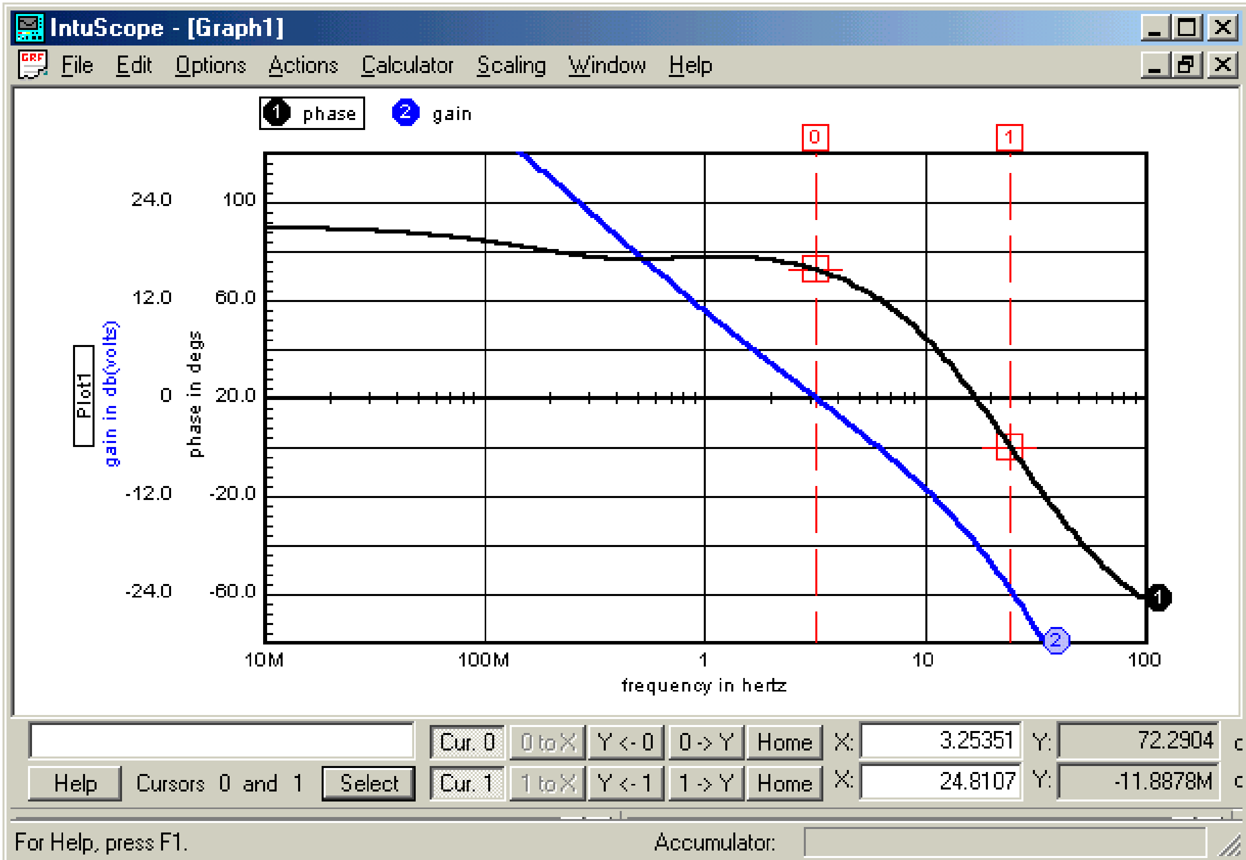

Figure 2.6. SPICE Plot of Loop Gain indicates 3.25Hz Bandwidth and 72.3° Phase Margin.

While the hardware step response in Figure 2.4 gives a rough indication of loop bandwidth and phase margin, it does not pave the way to optimization as the well as the loop gain plot in Figure 2.6. This graphic demonstrates that bandwidth is ultimately limited by sample frequency, and that proportional correction must exceed integral correction at the unity gain frequency. These types of valuable observations are easier to reach after working with analytical tools than trial and error methods.

For example, if this exercise were repeated without proportional gain the system step response would be on the verge of oscillation. This is because the lag of the integrator combined with the lag of the infrared circuit drives the phase of the loop dangerously close to 180° at the unity gain frequency. On the other hand, without integral gain the system would be stable but possess significant steady-state error. This is because DC loop gain is low with only the proportional term. If lower gains were used for both terms, the loop bandwidth would be reduced and the circuit response would be slow to changes. The proper combination of proportional and integral gain is essential for a fast, stable, and accurate response.

In this case, the frequency that proportional gain exceeds integral gain (f = 1 / (2Π•50K • 6.4µF) = 0.50Hz) is slightly after the corner frequency of the infrared circuit [f = 1 / (2Π•1MΩ • 0.47µF) = 0.34Hz) to maintain a single pole roll-off to unity loop gain. The proportional gain is selected to achieve a control loop bandwidth (3.25Hz) approximately one decade higher than the infrared circuit bandwidth (0.34Hz).

Figure 2.7 is the transient step response of the simulated infrared controller, which matches the hardware measurement in Figure 2.5 very well. Gains and time constants of the infrared hardware and software compensation were varied, and multiple simulations were successfully compared to hardware results to validate the proposed techniques.

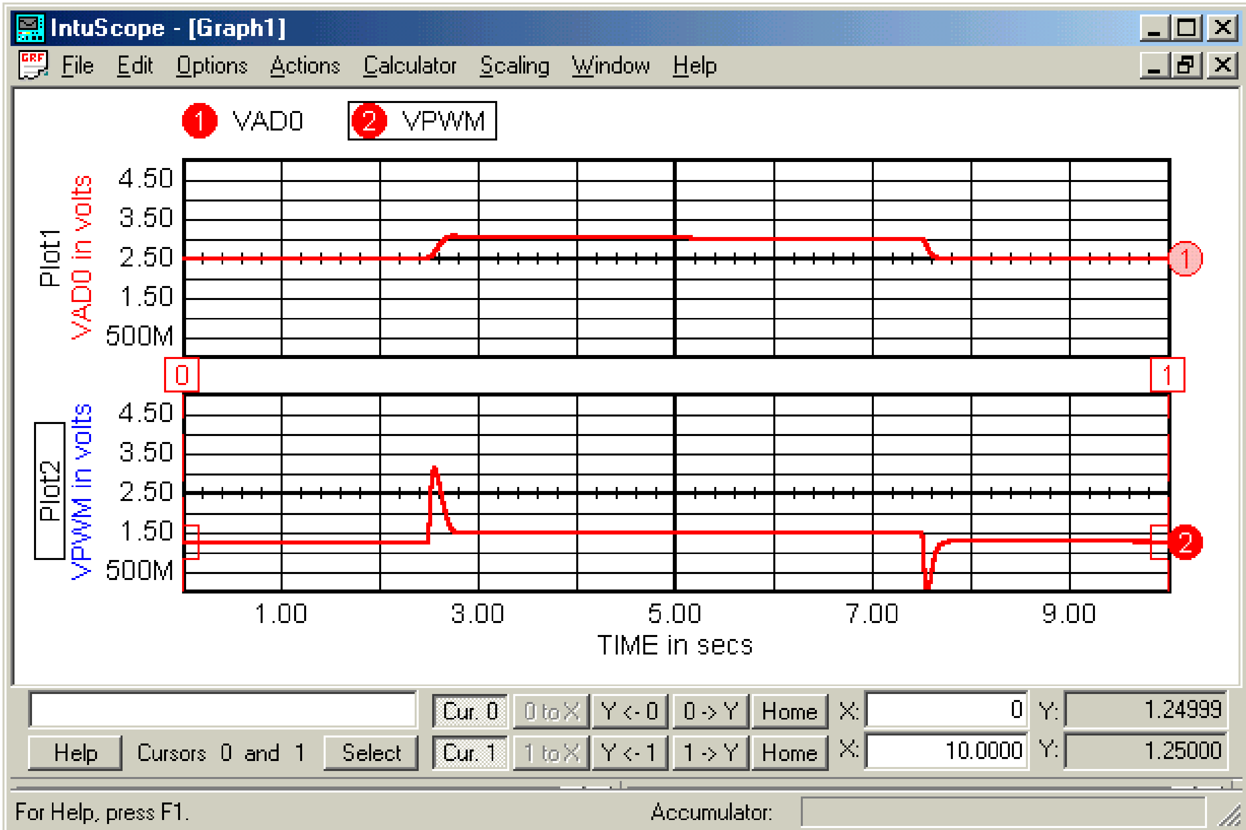

Figure 2.7. SPICE Plot of Transient Step Response matches Hardware Measurement.

These gains were chosen conservatively for a robust response despite changing conditions and production variations. Inspection of Figure 2.8 Bode Plot of loop gain reveals that a more aggressive design with higher bandwidth can be achieved by doubling both gains. Increasing KP to 10 (Rprop = 100KΩ) and KI to 0.4 (Cint = 3.2µF) results in an 8.1Hz loop bandwidth with 60° phase margin. This modification is experimentally confirming by passing a thin object through the infrared path shown in Figure 2.1 and noting a smaller AD0 disturbance in Figure 2.4 compared to the original gains of KP = 5 and KI = 0.2.

The advantages of transforming a system into its most fundamental elements for simulation are many. Greater intuitive understanding is achieved through both the transformation and simulation processes. Many people using varied software can share results, making more tools and application assistance available. Errors are less likely to go undetected when calculations are performed by multiple methods and compared. Better optimization with respect to performance, robustness, and parts selection is enabled with this enhanced understanding and availability of tools. Most importantly, decades of useful knowledge are put to work to benefit new and emerging technologies, which is essential for continued growth.

3.0 FlexController™ System-On-Chip Part FCIC010

FlexController™ is a single chip solution that combines powerful mixed-signal hardware with advanced real-time firmware. A versatile data interface enables multiple units to communicate with a PC through a fast convenient Active X Control. Industrial applications include control of power, thermal, motion, lighting, and flow, while educational applications include electronics, programming, simulation, and control systems.

Highlights:

16 I/O Total (9 Input / 7 Output)
5 Analog Inputs
1 Timer, 1 Counter, & 1 Clock
3 PWM Outputs for Power Control
EEPROM Reference Calibration
Preprogrammed functionality
Multi-Drop Serial Interface
Data Acquisition and Logging Capability
Fast Updates for Closed Loop Control
Free PC Software & Active X Control
Very Low Cost and Very Easy to Use

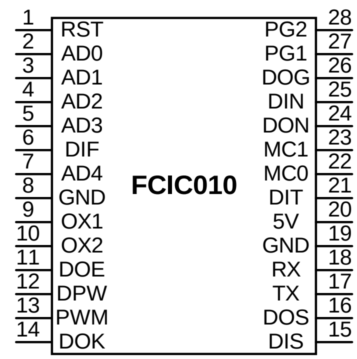

Figure 3.1. FCIC010
28 Pin DIP 0.3in Narrow Package

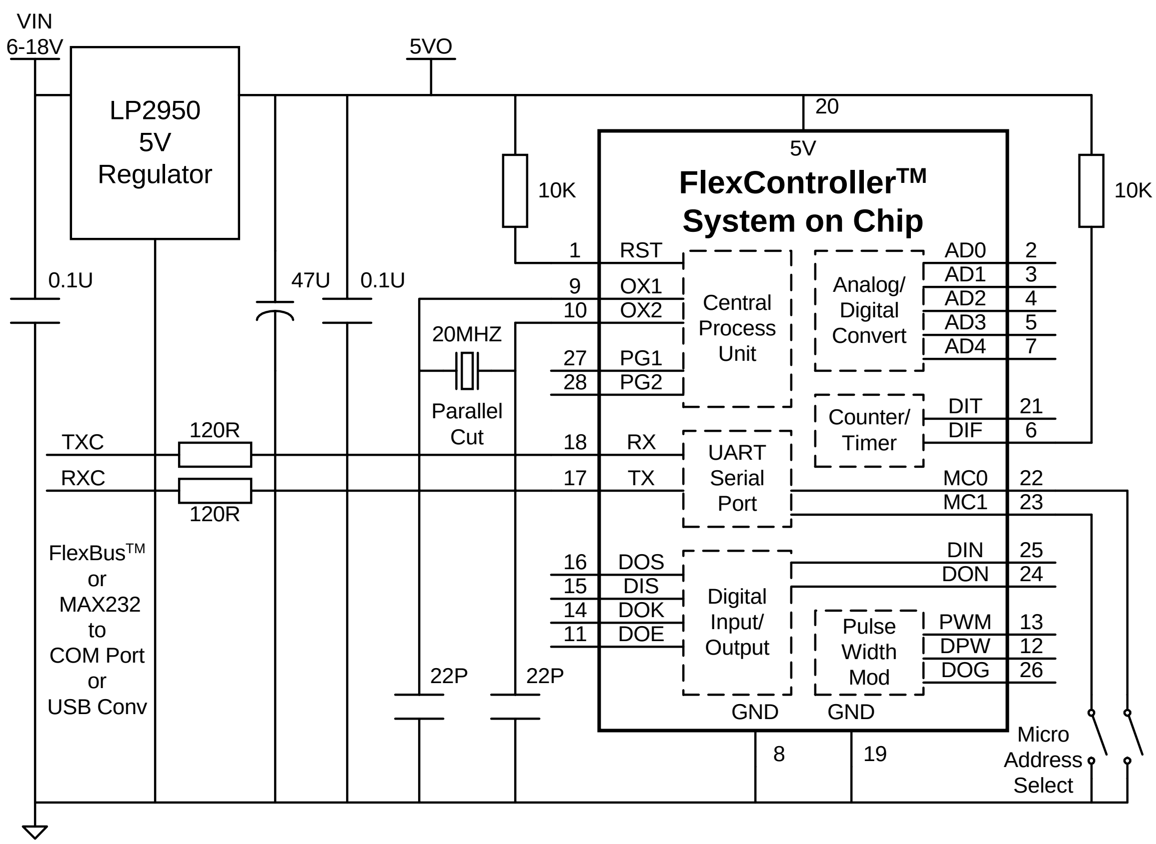

Figure 3.2. Typical FlexController™ Circuit

Pinout

Pin	Name	Description
1	RST	Reset Low (External Pull-Up Required)
2	AD0	Analog to Digital Converter
3	AD1	Analog to Digital Converter
4	AD2	Analog to Digital Converter
5	AD3	Analog to Digital Converter
6	DIF	Totalizing Counter Input (Schmitt Trigger)
7	AD4	Analog to Digital Converter
8	GND	Power Gnd
9	OX1	20Mhz Crystal (22pF Capacitor Required)
10	OX2	20Mhz Crystal (22pF Capacitor Required)
11	DOE	Digital Output
12	DPW	Software PWM
13	PWM	Pulse Width Modulator
14	DOK	Digital Output
15	DIS	Digital Input (Internal Pull-Up)
16	DOS	Digital Output
17	RX	Serial UART Receive
18	TX	Serial UART Transmit
19	GND	Power Gnd
20	5V	5V Power
21	DIT	Interval Timer Input (Schmitt Trigger with Internal Pull-Up)
22	MC0	Micro Address LSB (Internal Pull-Up)
23	MC1	MIcro Address MSB (Internal Pull-Up)
24	DON	Digital Output
25	DIN	Digital Input (Internal Pull-Up)
26	DOG	Software PWM
27	PG1	Programming Input (Reserved for future use)
28	PG2	Programming Input (Reserved for future use)

Figure 3.3. FlexController™ SOC Pinout

Specifications

Parameter	Specification
5V Power	5V ± 0.25V / 9mA Typical
Operating Temperature	-25C to 85C
ADC Resolution	10Bit Hardware / Software Enhanced to 12Bit (Over-sampled)
ADC Input Range	0 to 5V / External Source Resistance < 5Kohm
DIT Interval Timer	24Bit / 50uS to 13S Range / 800nS Res / Neg Edge Trig
DIF Totalizing Counter	24Bit / 16,777,216 Rollover / 1MHz Max Rep Rate / Neg Edge
CLK Running Clock	16Bit / 500uS Resolution / 32.768 Sec Rollover
PWM Output	20KHz Repetition Rate / 10Bit Pulse Resolution
DPW & DOG Outputs	125Hz Repetition Rate / 7Bit Pulse Resolution
RST Reset Pin	<0.8V (<750ohm to GND) > 10uS (Filtered)
Digital Inputs	CMOS Inputs (LO < 0.8V & HI > 2.4V, Pull-ups = 250uA)
Digital Outputs	CMOS Outputs (0 or 5V unloaded / 25mA Max)
EEPROM Storage	16Bit Vref Correction & 8 Character Alphanumeric ID String
Serial Update	38.4KBd / 12.8mS Update Typical to Write & Read All

Figure 3.4. FlexController™ SOC Performance Specifications

Applications

Figures 3.5 and 3.6 illustrate typical FlexController™ applications, while demonstrating the versatility and multi-functionality of a single chip. The high resolution and frequency of the PWM Output (Figure 3.5) is ideal for precisely regulating power delivered to actuators (M1) for desired operation. Feedback could be voltage (AD2), current (AD1), or speed measured as the time between pulses (DIT at lower speeds) or pulse count within a known interval (DIF at higher speeds).

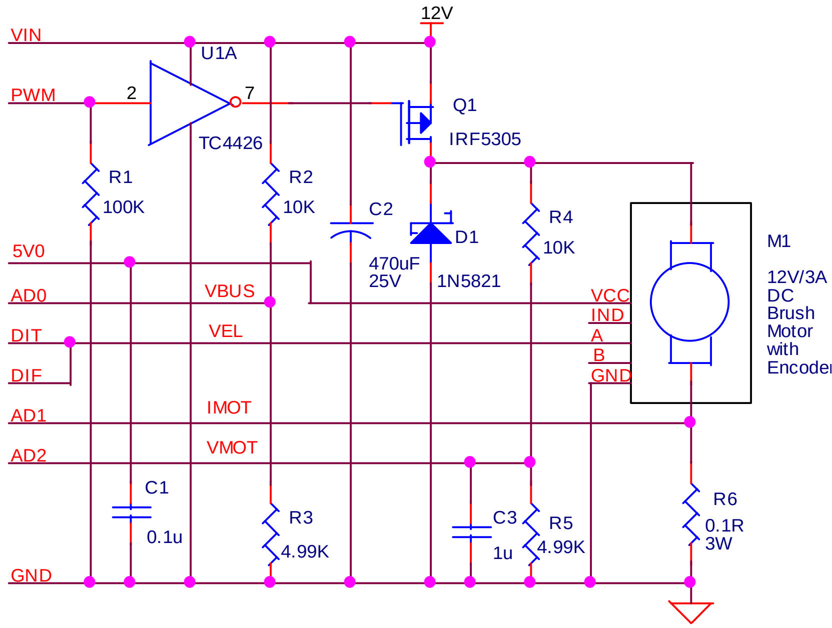

Figure 3.5. Motor Control Application

The DPW and DOG Outputs (Figure 3.6) are well suited for lighting (DPW/AD3) and thermal (DOG/AD4) applications as excellent reponse is achievable with simple yet efficient power devices. Additional features could be added such as undervoltage lockout or overvoltage shutdown (AD0) with I/O remaining for other functions.

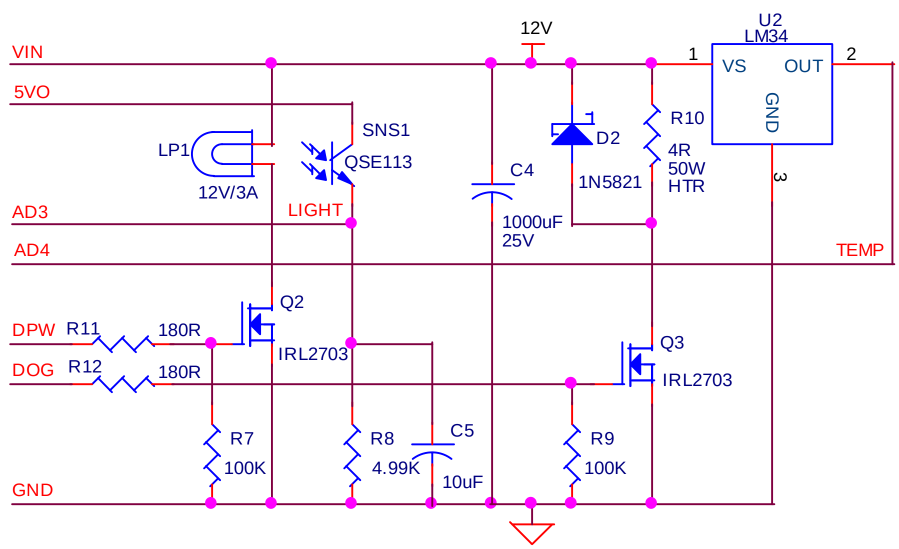

Figure 3.6. Lighting and Thermal Control Application

Low pass filtering the PWM Output yields a precise Digital to Analog Converter (DAC) as step changes are a function of exact time dependent ratios (crystal clock), as opposed to the approximate resistance ratios of integrated circuit converters.

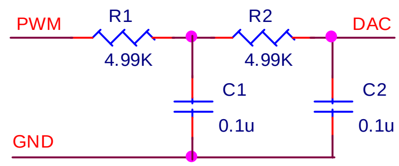

Figure 3.7. Low Pass Filter PWM for Precise Analog Output DAC

An op-amp can be added to the circuit of Figure 3.7 for buffering or gain. In additon, a transistor pass element may be used to convert the averaged PWM drive into a linear power amplifier. Power components with current and thermal limit circuitry on a heat sink should be used in this type of application for protection against damage from excessive load.

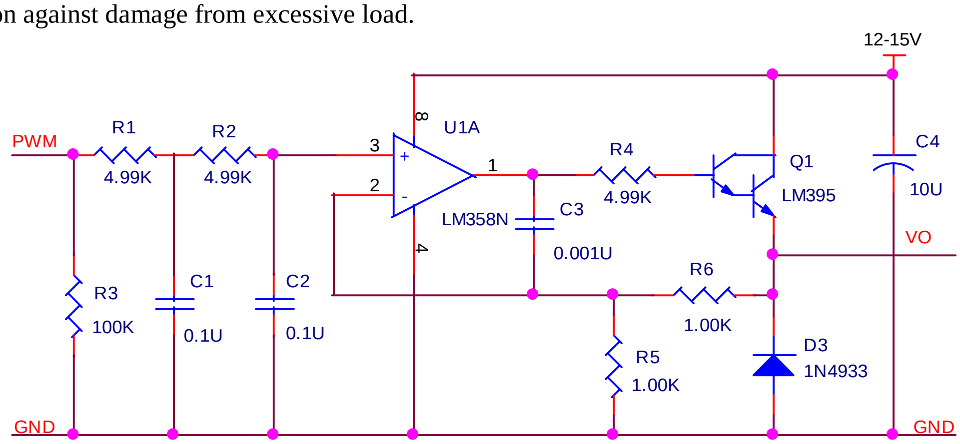

Figure 3.8. Digitally Controlled 10V/1A Linear Power Amplifier

Advantages

There are many advantages to the versatile, low-cost, easy-to-use FlexController™ SOC in comparison to other systems.

Other Approaches	FlexController™ SOC
Design from scratch	Tremendous time savings and improved performance
Dedicated application chips	Reapply familiar proven component in countless designs
Data acquisition boards	Active control superior to passive observance
Single Board Computers	Smaller, less expensive, easier to program
Real-time control systems	Extreme cost savings
Most PC-based tools	Free intuitive software
Other versatile low-cost easy-to-use SOC	None found to compare

Figure 3.9. Advantages of FlexController™ SOC over other approaches

Communication

The serial interface to FlexController™ may be through an industry standard MAX232 converter chip to a PC COM Port or USB to Serial Converter. However, FlexBus™ developed by Flextek has many advantages over other serial interfaces (covered in FlexBus™ section). The FTview™ Active X Control enables quick and easy custom programming of FlexController™ SOC (covered in FTview™ section).

FTview™ Active X enables simple PC communication with FlexController™ SOC but detailed serial protocol is provided for micro-to-micro communication in embedded applications. Two commands, P for PUT and Q for QUERY, write and read 8-bit micro registers. All commands are six-byte ASCII upper-case strings. Micro will not transmit (TX high impedance) unless it receives a valid QUERY command, then it will return binary data for the number of bytes requested, followed by a single-byte checksum.

Hardware Specification: 38.4KBaud, No Parity, 8 Bits, 1 Stop Bit, No Handshaking

Byte 1	CMD	ASCII Command	P or Q
Byte 2	MIC	Micro Address	0 - 3
Byte 3	ADRH	Register Address High Nibble	0 - F
Byte 4	ADRL	Register Address Low Nibble	0 - F
Byte 5	VALH	Register Value High Nibble	0 - F
Byte 6	VALL	Register Value Low Nibble	0 - F

Value (16*VALH+VALL) is new register content in PUT command
Value is number of consecutive registers in QUERY command starting with micro address

Example Data Transfers:
PUT 18h into Register 38 and 80h into Register 39 of Micro Address 0.
ASCII PC TX: P03818P03980
QUERY 2 registers starting with Address 38 of Micro 0 returning binary data followed by checksum byte.
ASCII PC TX: Q03802
Binary PC RX: 188098
Operational result is 15% (24/160) duty cycle in DPW and 80% (128/160) duty cycle in DOG.

Addr	Reg	Description
23	DIO	B7=DOS, B6=DOK, B5=DOE, B4=DON, B3=DIS, B2=DIN, B1=DIT, B0=DIF
24	AD0H	AD0 High Byte of 16Bit Over-sampled ADC Conversion
25	AD0L	AD0 Low Byte
26	AD1H	AD1 High Byte of 16Bit Over-sampled ADC Conversion
27	AD1L	AD1 Low Byte
28	AD2H	AD2 High Byte of 16Bit Over-sampled ADC Conversion
29	AD2L	AD2 Low Byte
2A	AD3H	AD3 High Byte of 16Bit Over-sampled ADC Conversion
2B	AD3L	AD3 Low Byte
2C	AD4H	AD4 High Byte of 16Bit Over-sampled ADC Conversion
2D	AD4L	AD4 Low Byte
2E	CLKH	CLK High Byte of 16Bit 500uS Step Continuous Timer
2F	CLKL	CLK Low Byte
30	DIF2	DIF High Byte of 24Bit Negative Edge Triggered Counter
31	DIF1	DIF Middle Byte
32	DIF0	DIF Low Byte
33	DIT2	DIT High Byte of 24Bit Negative Edge Triggered Timer
34	DIT1	DIT Middle Byte
35	DIT0	DIT Low Byte
36	PWMH	PWM High Byte of 10Bit Left Justified Hardware PWM
37	PWML	PWM Low Byte
38	DPW	DPW Value of 160 Step (7.3Bit) Right Justified Software PWM
39	DOG	DOG Value of 160 Step (7.3Bit) Right Justified Software PWM

4.0 FlexBus™ Interface Board — Part FCIF010

FlexBus™ isolated multi-drop serial bus connects multiple FlexController™ SOC directly to PC COM port or USB port with serial converter. The electrical isolation of the data bus eliminates line drops and ground noise from measurements, protects the PC from power transients, and enables operation at voltages not possible with other systems. Socket (28-pin 0.3in DIP) accommodates FlexController™ SOC or common Microchip PIC devices.

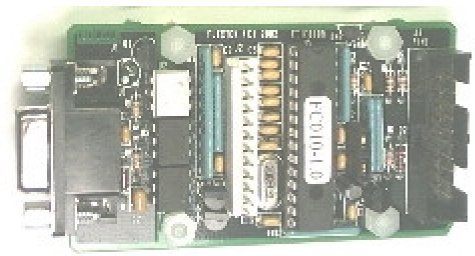

Figure 4.1. FlexBus™ Board (1.7in x3.0in PCB)

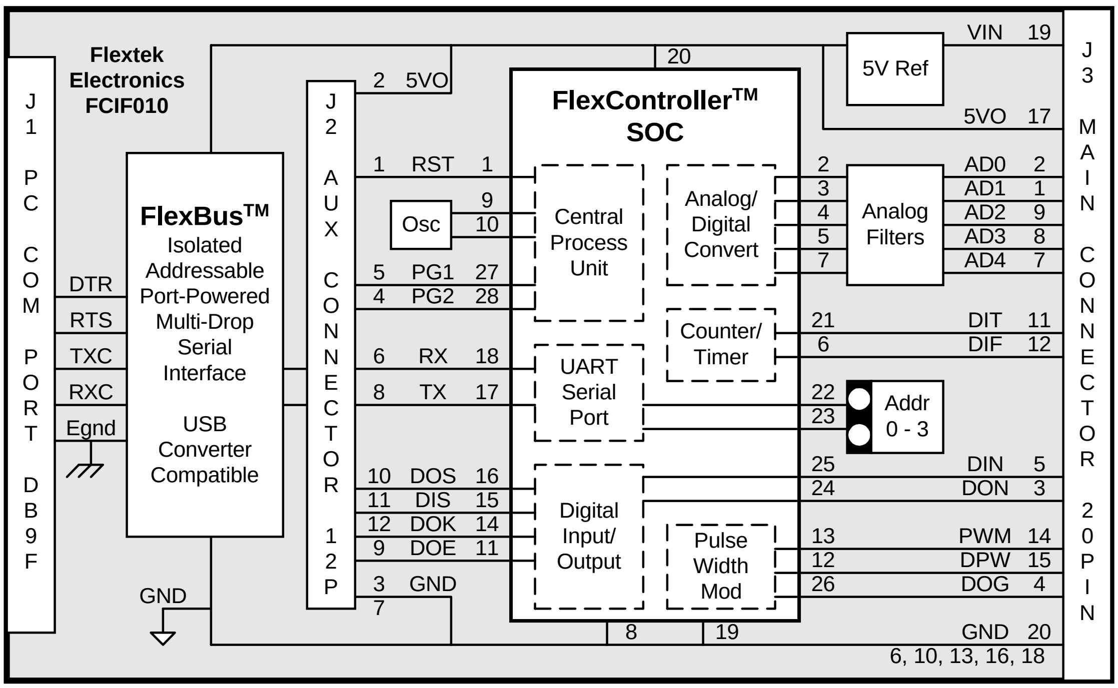

Figure 4. 2. FlexBus™ Board Block Diagram

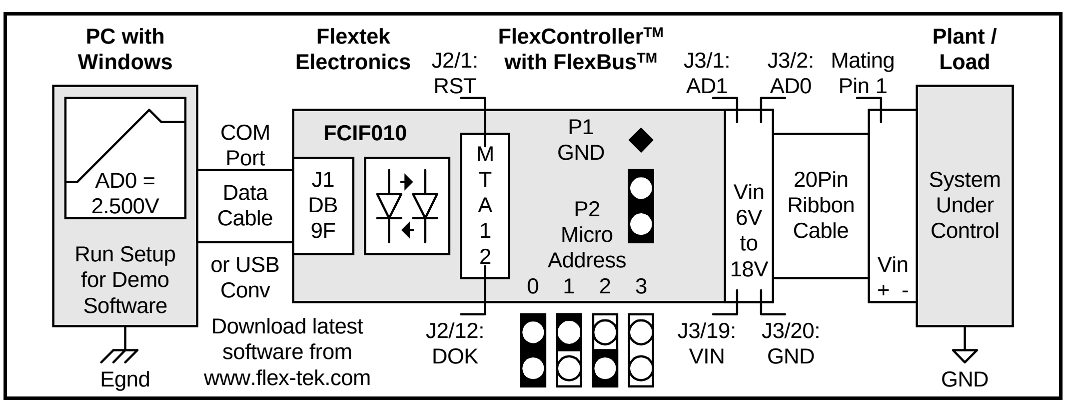

Figure 4.3. FlexController™ with FlexBus™ System Setup and Installation

Pinout

J1	DB9F	J2	12Pin MTA		J3	20Pin Ribbon		J3	20Pin Ribbon	
1	NC	**1**	RST	Reset In	**1**	AD1	Analog In	**11**	DIT	Timer In
2	RXC	**2**	5V0	Power Out	**2**	AD0	Analog In	**12**	DIF	Counter In
3	TXC	**3**	GND	Power Rtn	**3**	DON	Digital Out	**13**	GND	Power Rtn
4	DTR	**4**	PG2	Reserved	**4**	DOG	Pulse Out	**14**	PWM	Pulse Out
5	Egnd	**5**	PG1	Reserved	**5**	DIN	Digital In	**15**	DPW	Pulse Out
6	NC	**6**	RX	Comm In	**6**	GND	Power Rtn	**16**	GND	Power Rtn
7	RTS	**7**	GND	Power Rtn	**7**	AD4	Analog In	**17**	5VO	Power Out
8	NC	**8**	TX	Comm Out	**8**	AD3	Analog In	**18**	GND	Power Rtn
9	NC	**9**	DOE	Digital Out	**9**	AD2	Analog In	**19**	VIN	Power IN
		10	DOS	Digital Out	**10**	GND	Power Rtn	**20**	GND	Power Rtn
		11	DIS	Digital In						
		12	DOK	Digital Out						

Figure 4.4. FlexBus™ Board Pinout

Specification

Parameter	Specification
VIN Supply	6V to 18V
Supply Current	3mA Typical
5VO Reference Voltage	5V +/-25mV Typical
Operating Temperature	0 to 70C
Isolation (GND to Egnd)	1000V with Nylon Standoffs (50V with Aluminum Standoffs)
Serial Baud Rate	34.8KBd Maximum
TX/RX Series Resistance	120Ω
RTS Voltage Range	+5V to +12V
DTR Voltage Range	0V to –12V

Figure 4.5. FlexBus™ Board Specifications

Applications

The Flextek isolated port-powered multi-drop serial interface can communicate with up to four FlexController™ SOC as shown in Figure 4.6 (any combination of boards or chips), or other types of serial devices when the FlexController™ SOC is removed from its socket. Clear signal RTS in PC software for positive bias and set DTR for negative bias to the serial interface (PC RS232 drivers invert). Protect TX & RX pins of ICs with 120Ω resistors and do not exceed 38.4KBaud rate due to opto-coupler bandwidth limitations.

Example applications include the individual charge control of each battery in a series stack , troubleshooting a negative voltage supply, or commanding high side power devices, all with minimal cost and effort. Simply daisy chain data cable with multiple DB9 ribbon cable connectors or a DB9 serial "Y" splitter cable. The serial data rate, buffer length, and protocol have been selected to be compatible with common USB to serial converters for maxium flexibility.

FlexBus™ may also be used in embedded applications for micro-to-micro isolated multi-drop communication. Figure 4.7 demonstrates stand-alone operation where FlexController™ with FlexBus™ serve as intelligent peripherals to a user programmed micro. FlexController™ SOC documentation explains the simple two-command serial protocol for embedded applications. The versatile FlexBus™ interface may be powered by ±12V for PC applications as in Figure 4.6, or a single 5V supply for the embedded application shown in Figure 4.7.

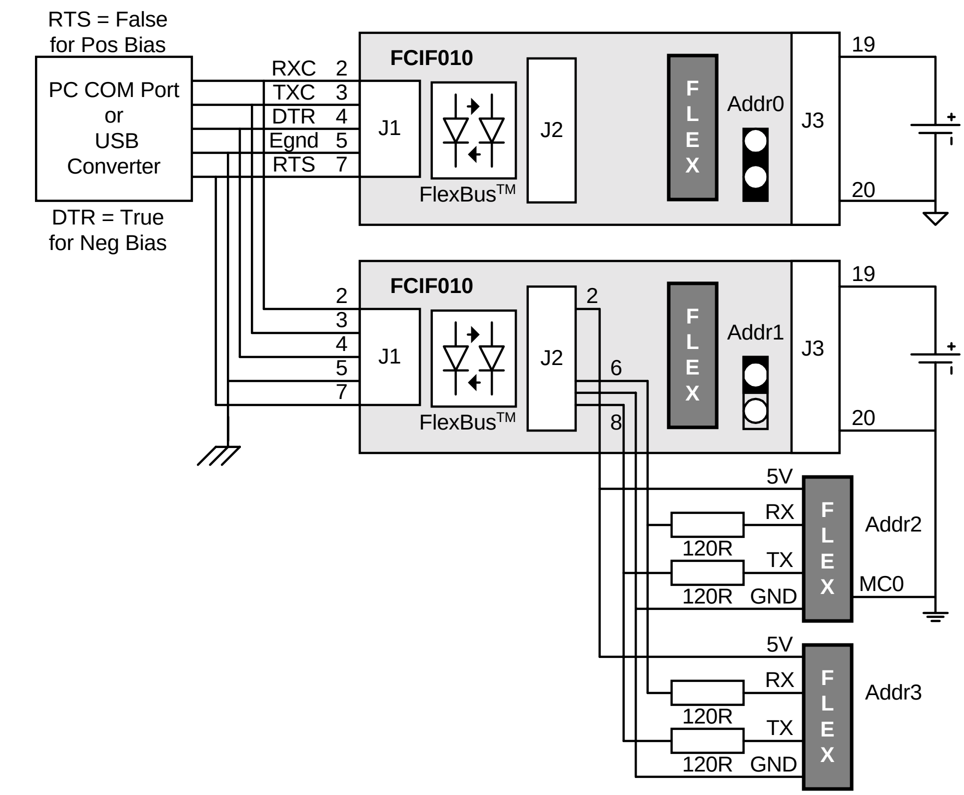

Figure 4.6. FlexBus™ Enables Isolated Multi-Drop Communication

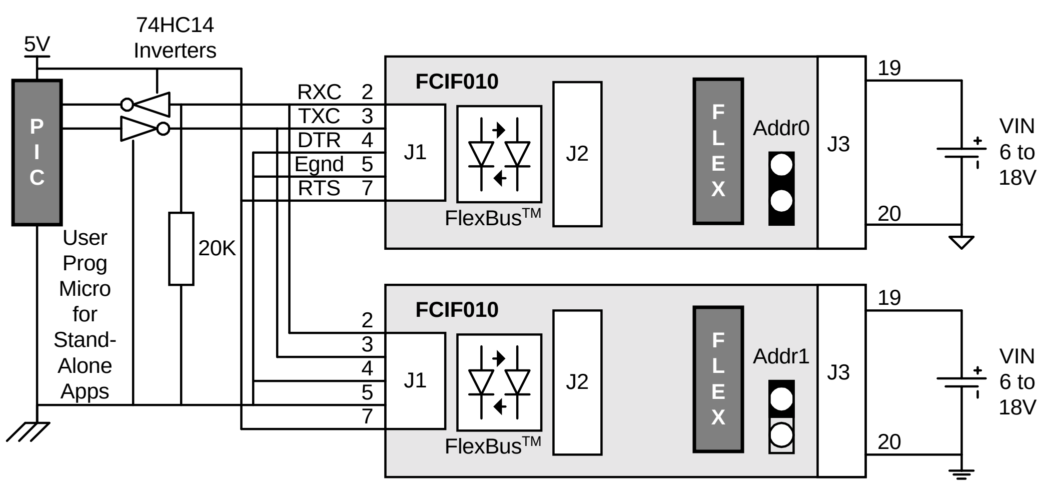

Figure 4.7. FlexBus™ Communicates with other Micros in Embedded Applications

Advantages

The addressable multi-drop data bus allows additional boards and chips to be included as system demands grow, without the expense of data expansion modules. The electrical isolation of the bus eliminates line drops and ground noise from measurements, protects PC from power transients, and enables tasks not possible with other systems. FlexBus™ is superior to other serial interfaces for overall effectiveness and value in interactive control applications.

FlexBus^{TM}	USB
Electrical isolation is robust, safe, and clean	Risk ground loops, transients, & noise
Device returns at different voltages	Device returns at earth ground only
Four devices per port	One device per port
Cable length >25ft capability	Cable length <10ft limitation
Works with any PC	Requires newer operating systems
Proven reliability	Problems and compatibility issues
Low cost	Moderate cost
PC or embedded applications	Only PC applications
Wide variety of compatible micros	Limited selection of compatible micros
Low power (3mA) with Insta-Standby^{TM}	Power hungry (25mA) with manual standby
No software overhead	Extensive software overhead
Interrupt driven	Software polled
Fast efficient interactive data transfer	High data rates for large buffered data
12.8mS typical Read/Write update interval	50mS typical Read/Write update interval
Superior in control applications	Better for cameras and scanners

Figure 4.8. FlexBus^{TM} vs. USB Comparison for Control Applications

Interface Details

The FlexBus^{TM} Board has many features for robust operation, including ground plane, protection diodes, current-limit resistors, filter capacitors, and electrical isolation. Details follow to ensure proper connection of interface circuitry.

RESISTORS: Output resistors limit current and input filters attenuate noise. The RST and DIF pins have pull-up resistors, while DIT and DIN have 250uA pull-ups internal to the IC. There is a pull-down resistor on the DON pin to keep it low during power-up and reset (useful to disable external devices). Digital pins on J2 connector are not protected.

5VO: VIN can be left open to power 5VO externally if on-board reference is not desired. Voltage limited by 5.6V zener diode D2 (1N5232B).

RST: Filtered to prevent inadvertent reset due to noise. Output pins of chip are high impedance during reset then cleared low.

DIF: Very short (<200nS) pulses will be attenuated by the DIF filter and will not trip the counter. Allow longer (>2uS) for a high state to charge the filter capacitor if an open drain circuit is driving the DIF pin (charged by pull-up resistor only).

ADC: Low source impedance is recommended (<5Kohm) for best accuracy in consideration of ADC leakage current. Converter is over-sampled and software processed to achieve increased resolution , however ultimate accuracy is limited by hardware performance

CONNECTORS: The locking header of J3 provides a quick secure connection in production, and the plastic shell may be pulled away to give convenient access for test leads during prototyping.

Part Number	Manufacturer	Description
AK131-2	Assman	DB9F to DB9M 6ft Data Cable
M3AAA-2006J-ND	Digikey	20pin Socket to Socket 6in Ribbon Cable
103308-5	AMP	20pin PCB Header
1-644563-2	AMP	12pin MTA Receptacle

Figure 4.9. FlexBus^{TM} Board Interconnection Components

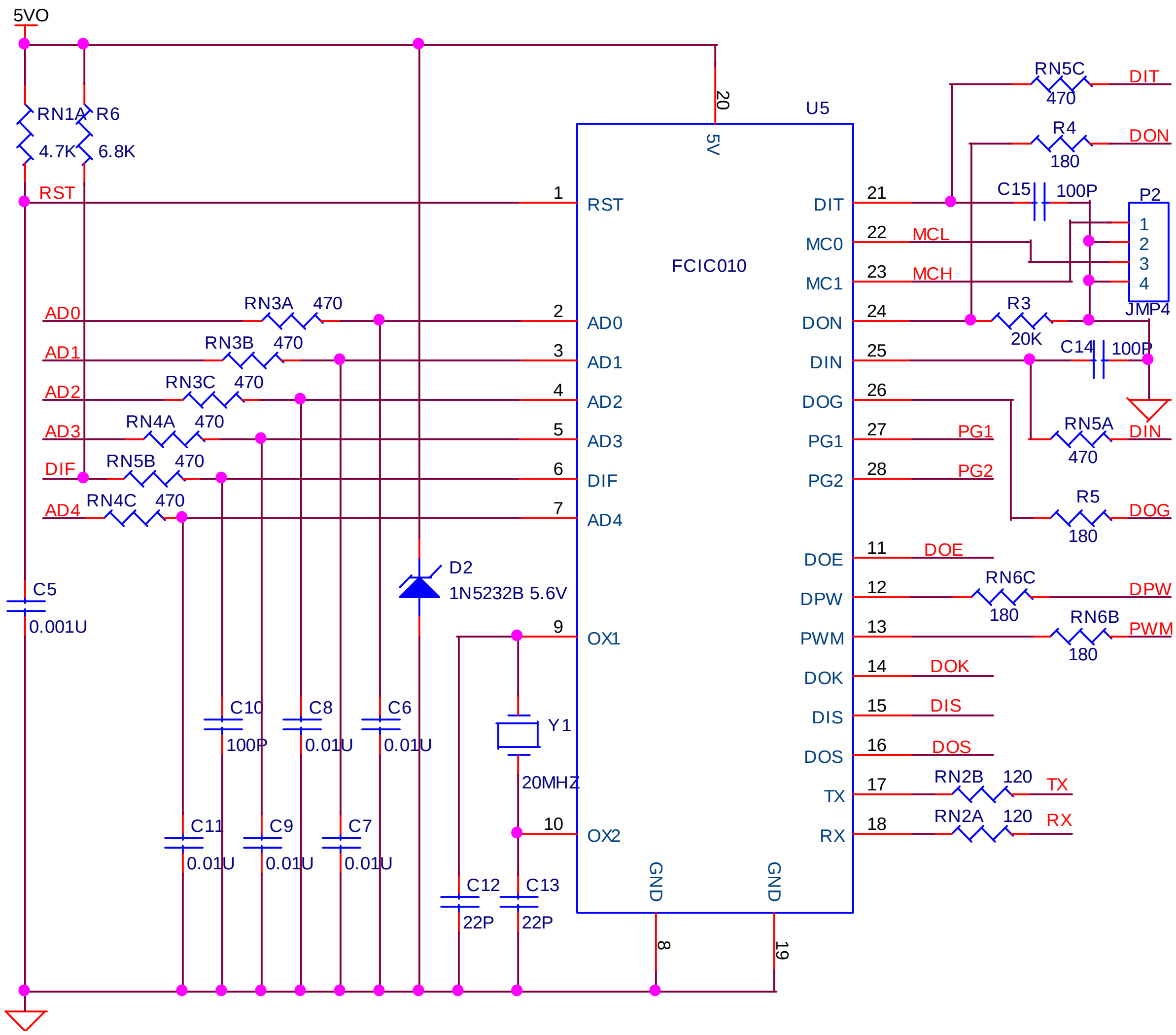

Figure 4.10. FlexBus™ Board Partial Schematic Illustrating Connection Interface Circuitry

DIMENSIONS: 4.11 lists the physical dimensions of the FlexBus™ Board including mounting holes and connector pins. This information is useful for custom enclosures or mating hardware. Allocate space left of board for serial data cable.

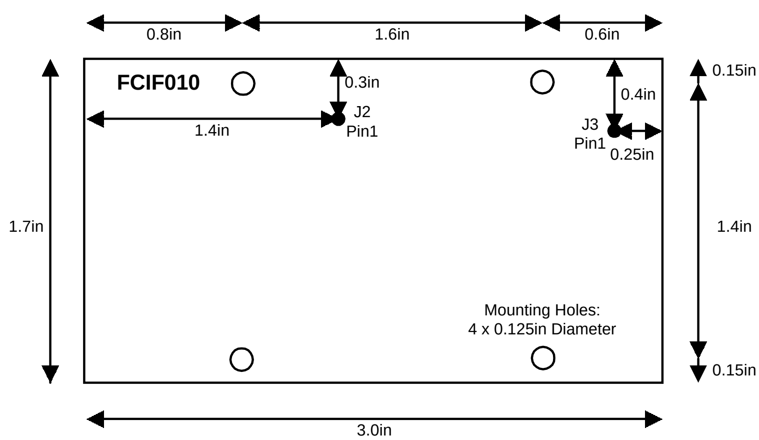

Figure 4.11 FlexBus™ Board Dimensions with Mounting Holes and Connector Pins

5.0 MultiDriver™ Power Board — Part FCMD010

MultiDriver™ Power Board combines FlexController™ SOC and FlexBus™ Interface with versatile power stage to save valuable time in satisfying challenging control applications (control chip and serial bus covered in separate sections of manual):

Power
- 12V to 48V Power Supply Range
- 6A Half-Bridge or 3A Full-Bridge Driver
- Current Limit, Thermal Shutdown, Bypass Capacitance
- 5V Precision Regulator

Control
- FlexController™ SOC Socket (Chip Sold Separately)
- 20MHz Oscillator, 5V Reference, Address Select
- FlexBus™ Isolated Multi-Drop Serial Interface
- PIC Micro Compatible

Board
- Convenient Screw Terminal Connectors
- Physical Dimensions of 2.3in x 3.0in x 0.9in
- Split Ground Plane, Noise Filters, Current Limit Resistors, Protection Diode
- Connection for Microchip ICD Development

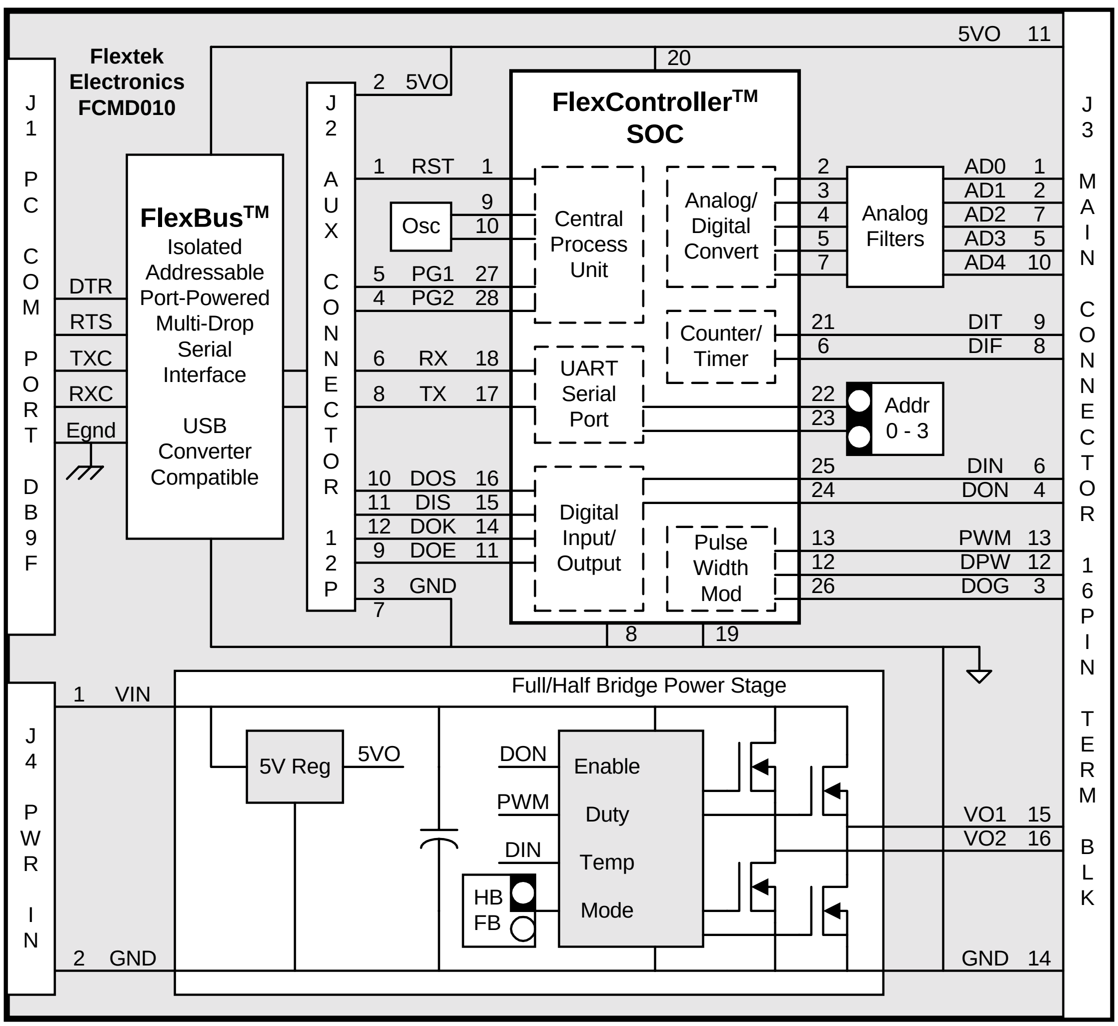

Figure 5.1. MultiDriver™ Board Diagram

Specification

Parameter	Specification
VIN Power	12V to 48V
Isolation (GND to Egnd)	1000V with Nylon Standoffs (50V with Aluminum Standoffs)
Operating Temperature	0 to 70C Ambient
VO1 & VO2 Current *	3A Each Full-Bridge / 6A Parallel Half-Bridge
Bridge Thermal	145C Warning / 170C Shutdown
5VO Reference Voltage	5V +/-25mV Typical (16Bit Correction Factor in EEPROM)
Serial Update	38.4KBd / 12.8mS Update Typical to Write & Read
TX/RX Series Resistance	120ohm
RTS Voltage Range	+5V to +12V
DTR Voltage Range	0V to –12V

* Derate to 2A Full-Bridge and 4A Half-Bridge without Fan or Heat Sink (Aavid 590102B)

Figure 5.2. MultiDriver™ Board Specification

Pinout

J1	DB9F	J2	12Pin MTA		J3	16Pin Term Block		J4	2Pin Term Block	
1	NC	**1**	RST	Reset In	**1**	AD0	Analog In	**1**	VIN	Power In
2	RXC	**2**	5V0	Power Out	**2**	AD1	Analog In	**2**	GND	Power Rtn
3	TXC	**3**	GND	Power Rtn	**3**	DOG	Pulse Out			
4	DTR	**4**	PG2	Reserved	**4**	DON	Digital Out			
5	Egnd	**5**	PG1	Reserved	**5**	AD3	Analog In			
6	NC	**6**	RX	Comm In	**6**	DIN	Digital In			
7	RTS	**7**	GND	Power Rtn	**7**	AD2	Analog In			
8	NC	**8**	TX	Comm Out	**8**	DIF	Counter In			
9	NC	**9**	DOE	Digital Out	**9**	DIT	Timer In			
		10	DOS	Digital Out	**10**	AD4	Analog In			
		11	DIS	Digital In	**11**	5VO	Power Out			
		12	DOK	Digital Out	**12**	DPW	Pulse Out			
					13	PWM	Pulse Out			
					14	GND	Power Rtn			
					15	VO1	Power Out			
					16	VO2	Power Out			

Figure 5.3. MultiDriver™ Board Pinout

DON: Set high to enable power stage. Both VO1 and VO2 high impedance when DON low.
DIN: Pulled low by power stage thermal warning flag prior to driver shutdown.
PWM: Sets state of bridge driver when enabled. May be driven by external logic level if control chip removed and DON pulled high to 5V.
HB/FB: Select power mode jumper for Half-Bridge (HB) or Full-Bridge (FB) on Jumper P3.
Half –Bridge: Both power outputs VO1 and VO2 are switched in phase with the PWM command. Load is connected between the parallel connected outputs and ground for 6A unipolar drive.
Full-Bridge: VO1 is switched in phase with PWM while VO2 is out of phase. Load is connected from VO1 to VO2 for 3A bipolar applications. Zero power occurs at 50% duty cycle in anti-phase switching.
Caution: Do not connect VO1 to VO2 when in Full-Bridge mode.

NOTE: Ensure that DON is set to enable power stage.

Applications

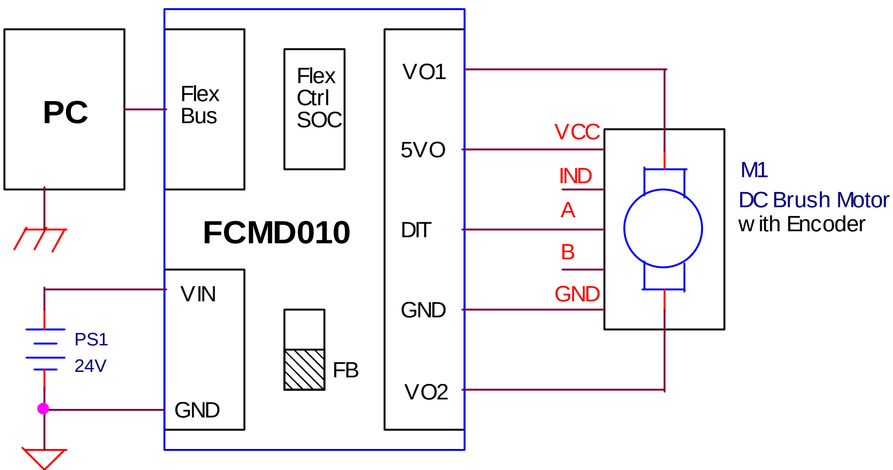

Figure 5.4. MultiDriver™ for Bi-Directional Motor Speed Control
(FlexController™ SOC with Full-Bridge Driver)

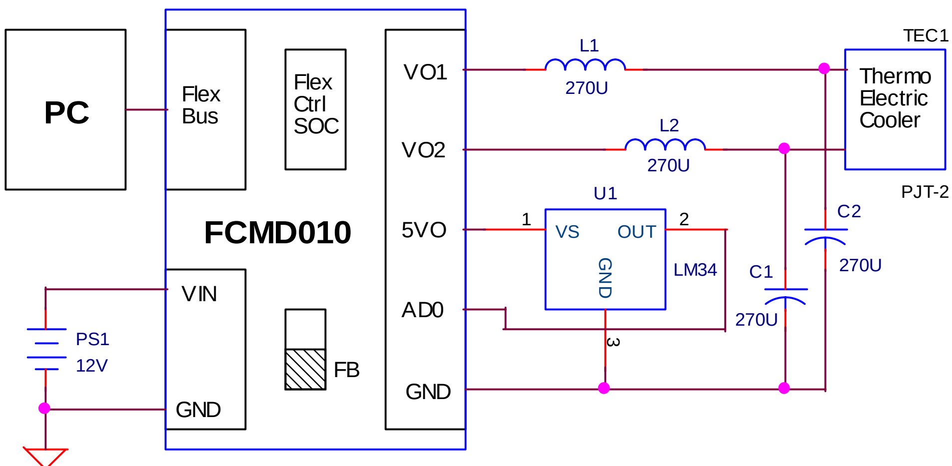

Figure 5.5. MultiDriver™ for Bi-Directional Thermal Control
(FlexController™ SOC with Full-Bridge Driver)

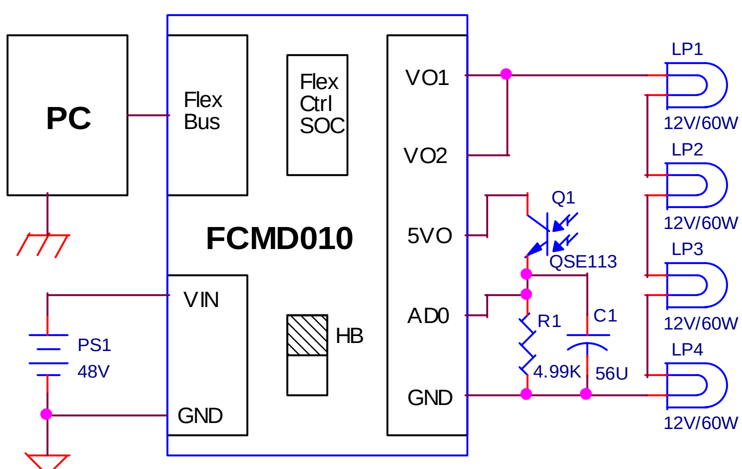

Figure 5.6. MultiDriver™ for High Power Light Control
(FlexController™ SOC with Half-Bridge Driver)

NOTE: Ensure that DON is set to enable power stage.

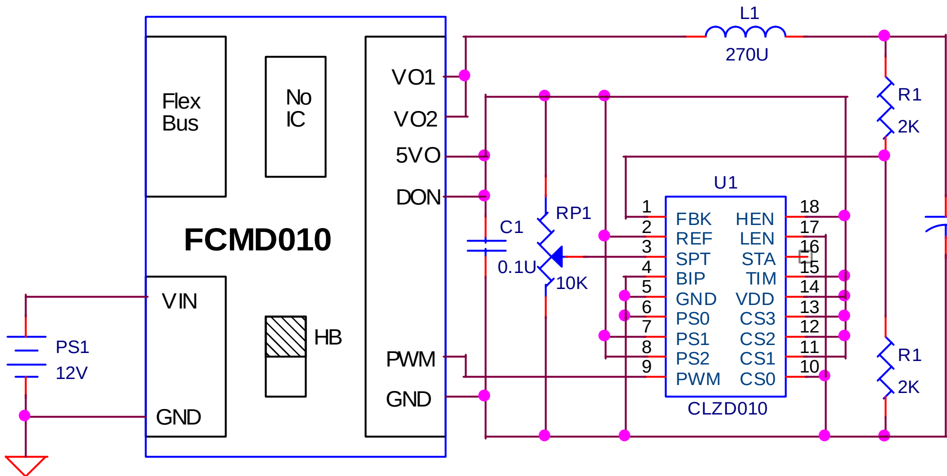

Figure 5.7. MultiDriver™ for DC/DC Converter Prototype
(External CLOZD™ Chip with Half-Bridge Driver – Control Socket on Board Empty)

The versatile MultiDriver™ board can be used for PC-Based Half-Bridge or Full-Bridge applications in power, thermal, motion and lighting, as well as embedded stand-alone control applications.

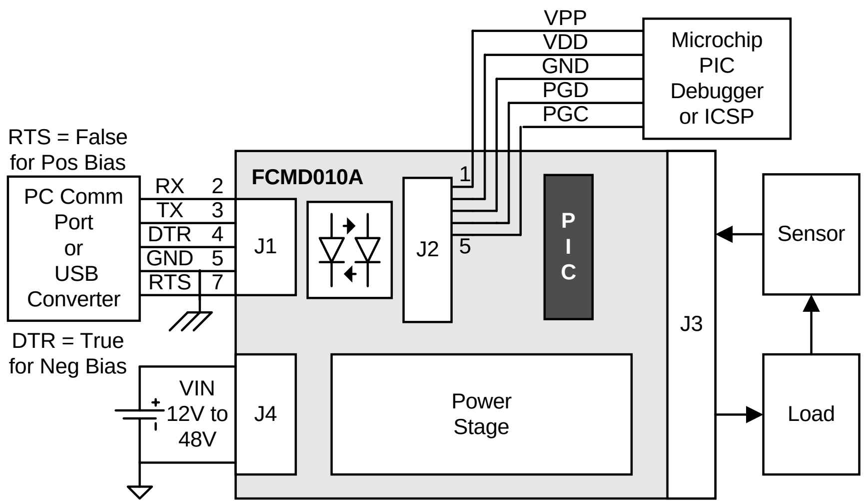

Figure 5.8. MultiDriver™ for Custom PIC Micro Development

FlexController™ chips use the same pinout as Microchip devices (PIC18F2320 and PIC16F873), so MultiDriver™ adds isolated interface, regulator, crystal, filters, connectors, and power stage to custom PIC applications. Flash pins are brought out to J2 for in-circuit programming and debugging with Microchip ICD.

NOTE: Ensure that DON is set to enable power stage.

6.0 FTview™ Software Part FTview10

FTVdemo is a free program that runs on any PC running Windows 95 or higher. It provides a simple yet effective interface to FlexController™. It is written in MS Visual Basic with the Active X Control FTview™. This simple control enables a wide variety of custom programs to be created quickly and easily.

FTVdemo can be downloaded from the Flextek Electronics web site at www.flex-tek.com and installed by running by its Setup program. FTview™ is included in the download for royalty-free creation of application programs that interface with FlexController™.

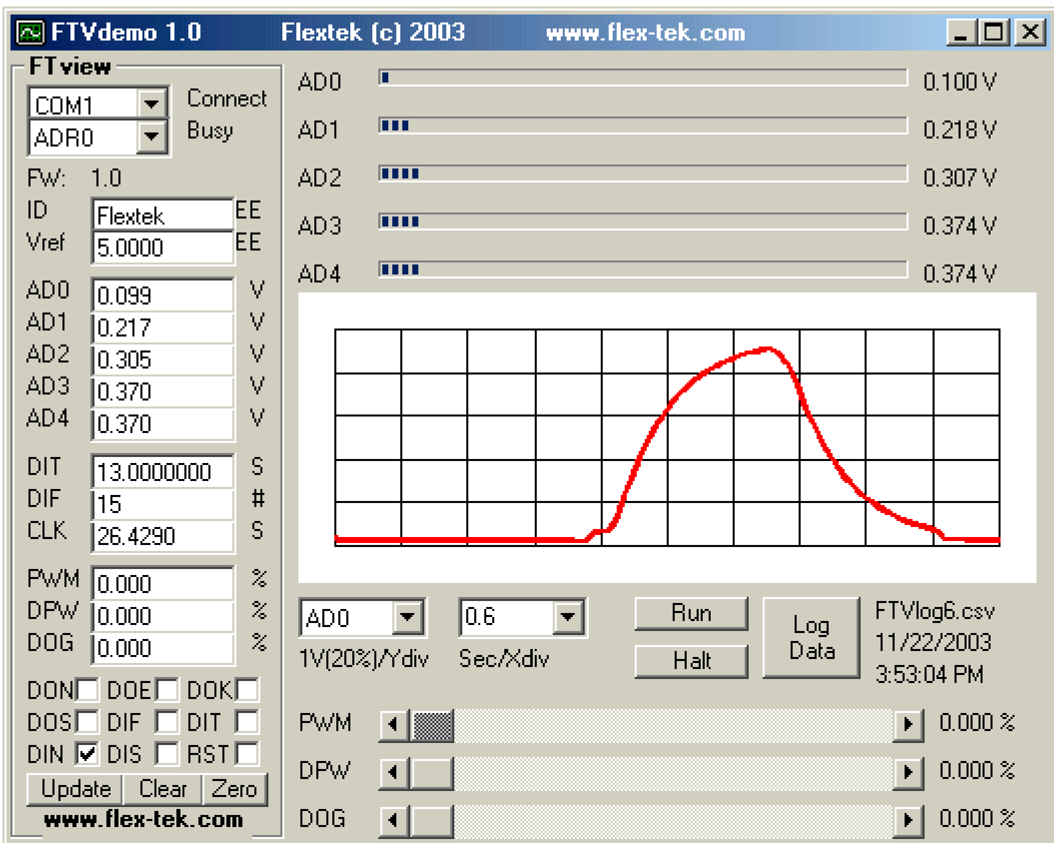

Figure 6.1. FTVdemo Program features Intuitive Controls with Plotting and Data Logging

Apply power to FlexBus™ Board with FlexController™ SOC and run the FTVdemo program. Select the PC COM Port and micro address from the FTview™ Control (defaults are COM1 and ADR0). Data will automatically update in the display window, including a user-selected plot.

The “Log Data” button will save the last 100 data points of all I/O values at the selected timing interval (6S, 60S, 600S full scale). The file will be assigned the next available name of the “FTVlog*.csv” format saved in the same directory as the application. These files can be opened by MS Excel for data display and manipulation, and should be moved before removal of the software application.

The heart of the FTVdemo program is the FTview™ Control that can be used manually (mouse and keyboard) or through program control. For example, type the PWM duty cycle value directly into the FTview™ Control or select it through code control of the horizontal scroll bar (FTView1.PWMduty = Hscroll1.Value). The program can address four different micros through one serial port but a single program should be run at any time from each serial port.

Education Example

Another program included in the FTVdemo download is FTVedkit. It interfaces with the Flextek Education Kit and illustrates programming techniques for closed-loop control. The kit hardware contains an Infrared Emitter and Sensor Board for a low-cost valuable experiment with an intuitive "touch-and-feel" for electronic control.

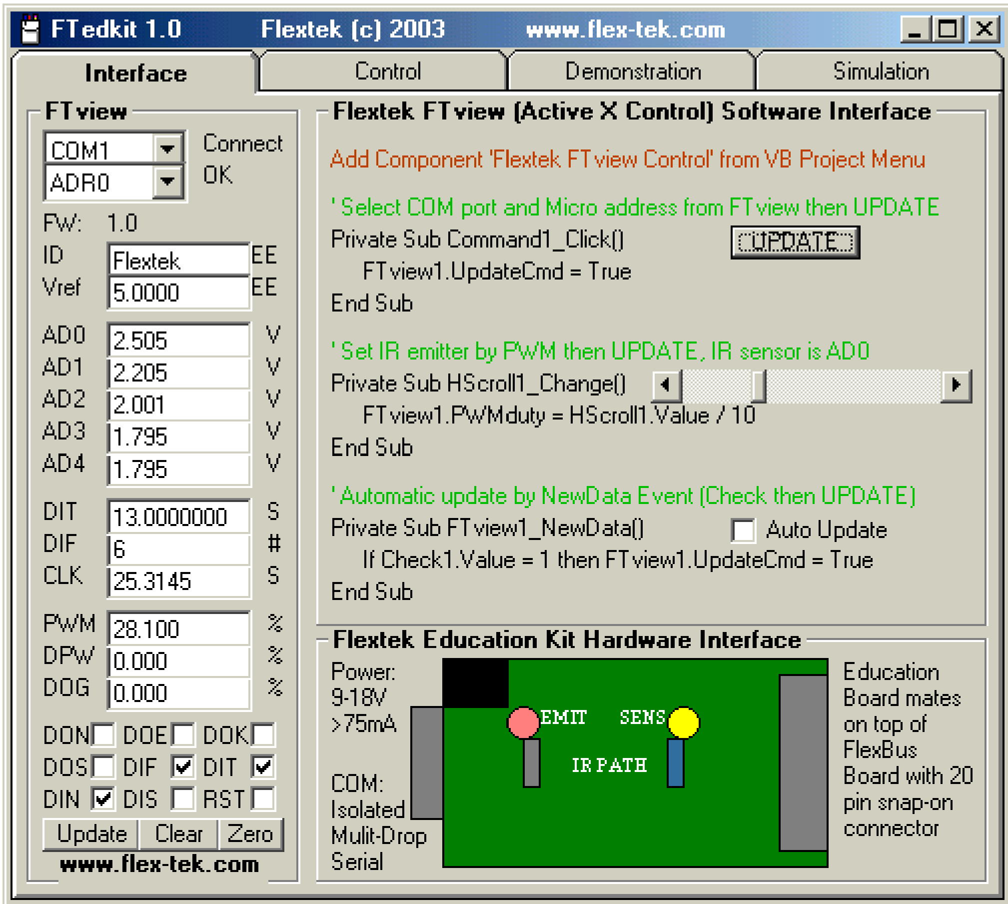

Figure 6.2. FTedkit Program illustrates Custom Programming and Closed-Loop Control

This application guides the user through the customization process by example. First load the Flextek FTview™ Control from the Components tab of the VB Project menu, then place it on a form and run the program. Select the PC COM Port and FlexController™ micro address from the control then click Update. This writes any user-modified registers then reads all values to be displayed in the control text boxes. Update can also be commanded through code control (FTview1.UpdateCmd = True).

Generate a timed loop by commanding a new micro Update every time the last Update completes and fires a NewData event. Code can be added to this loop to automate a desired task. This process is used in the FTVdemo program for continuous parameter updates, and in the Education Kit for Proportional-Integral control.

Custom Programming

Figure 6.3 further illustrates custom software development with FTview™ by writing a complete program from scratch in Visual Basic. This example provides constant speed control of a motor despite variations in motor load or power supply voltage. Refer to FlexController™ documentation for motor and driver schematic.

```
' FlexController Closed-Loop Speed Control of Motor with Encoder
' FTview reads DIT encoder period then writes PWM drive
Dim Setpt, KI, CPR As Single

Private Sub Form_Load()                   ' Initialization
   Setpt = 2000                           ' Desired RPM Speed
   KI = 0.003                             ' Loop Integrator Gain
   CPR = 500                              ' Codes Per Rev of Encoder
   FTview1.CommPort = 1                   ' Open Serial COM Port 1
   FTview1.UpdateCmd = True               ' Update control data
End Sub

Private Sub FTview1_NewData()             'Executes when Update done
   RPM = 60 * (1 / FTview1.DITinterval) / CPR   ' Read DIT encoder period and calculate RPM
   FTview1.PWMduty = FTview1.PWMduty + KI * (Setpt - RPM) ' Integrate RPM error for new PWM
   FTview1.UpdateCmd = True               ' Update with new PWM and read next timer value
End Sub                                   ' Sub loops at Update rate
```

Figure 6.3. Visual Basic Example using FTview™ for Closed-Loop Control

Use the following procedure to write your own program:
1) Download latest software from www.flex-tek.com and run SETUP to install
2) Open Visual Basic and select Components from Project menu
3) Select Flextek FTview Control then OK to load
4) Click FTview in Toolbox then draw control on form
5) Double click form to add initialization code and double click FTview to add closed-loop code
6) Run to test program then compile to automate your control application

Advantages

The powerful yet simple FTview™ software command set allows a full range of custom programs to be created quickly and easily. These intuitive commands replace countless tedious tasks including data port configuration, micro address decoding, and data formatting. The micro firmware contains digital filtering, error detection, as well as synchronization of hardware events through a real-time embedded kernel. The hardware incorporates analog filtering, protection circuitry, and innovative data interface. The combined features, performance, and cost of Flextek products make laborious design-from-scratch and expensive turnkey systems both obsolete.

Function Generator and Data Logger Example

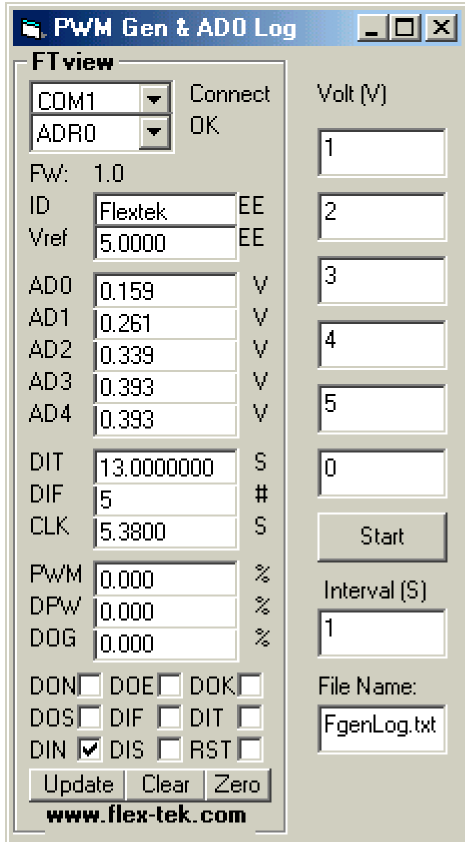

Figure 6.4. FTview™ in VB for PWM Function Generator and AD0 Data Logger

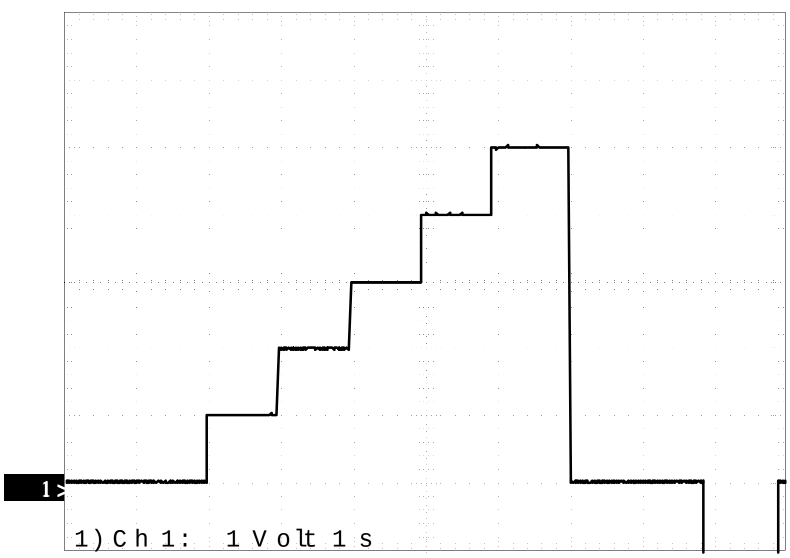

Figure 6.5. Scope trace of Filtered PWM (Averaged)

VB Code for PWM Generator and AD0 Logger

The code was written in Visual Basic with FTview™ to set the PWM of FlexController™ SOC to the voltage levels specified in Figure 6.4 text boxes at the desired time interval, then save the returned AD0 values in a text file. The sequence is executed once when the "Start" command button is clicked. This is accomplished by setting a timer to write the PWM and update the SOC, then read and save AD0 when the data is ready.

```
Dim Stp As Single                              ' Step Counter

Private Sub Form_Load()                        ' Initialization
   FTview1.CommPort = 1                       ' Serial COM Port 1
   FTview1.UpdateCmd = True                    ' Update Micro
End Sub

Private Sub Command1_Click()                 ' Start timer sequence
   Timer1.Interval = 1000 * Text7.Text        ' Time step in mS
   Timer1.Enabled = True                       ' Enable Timer
   Open Text8.Text For Output As 1            'Open file specified
End Sub

Private Sub Timer1_Timer()                  ' Executes at timer interval
   Stp = Stp + 1                              ' Increment step counter
   Select Case Stp                       ' Set PWM to value of next text box
      Case 1: FTview1.PWMduty = 100 * Text1.Text / 5
      Case 2: FTview1.PWMduty = 100 * Text2.Text / 5
      Case 3: FTview1.PWMduty = 100 * Text3.Text / 5
      Case 4: FTview1.PWMduty = 100 * Text4.Text / 5
      Case 5: FTview1.PWMduty = 100 * Text5.Text / 5
      Case 6: FTview1.PWMduty = 100 * Text6.Text / 5
   End Select
   FTview1.UpdateCmd = True         ' Update Micro with new PWM
End Sub

Private Sub FTview1_NewData()         ' Executes when Data Ready
   ' Write Step, PWM, and AD0 to file if ready
   PWM = FTview1.PWMduty: AD0 = FTview1.AD0volt
   If Stp <> 0 Then Write #1, Stp, PWM, AD0
   ' Disable timer and close file when done
   If Stp >= 6 Or FTview1.MicroStatus = "None" Then
      Stp = 0
      Timer1.Enabled = False
      Close #1
   End If
End Sub
```

Command Set

Property:	**AD0volt**
Description:	Returns voltage at AD0 input.
Example:	Vbus = FTview1.AD0volt * 3
Comment:	AD0 to AD4 operate identically.

Property:	**ClearCmd**
Description:	Clears (empty) all text boxes so next Update reads all micro registers (no values written).
Example:	FTview1.ClearCmd = True
Comment:	Failed UpdateCmd (no micro responded) issues ClearCmd.

Property:	**CLKtime**
Description:	Returns running clock synchronized to Update of micro in seconds.
Example:	NewTime = FTview1.CLKtime
Comment:	Useful in control loop or counter to frequency conversion.

Property:	**CommPort**
Description:	Opens selected COM port for serial communication. Ports 1 to 8 available in software.
Example:	FTview1. CommPort = 1
Comment:	Failure opening port returns COM OFF (CommPort = 0 for port closed).

Property:	**CommStatus**
Description:	Returns "Offline" or "Connect" dependent on COM port status.
Example:	ComStat = FTview1.CommStatus
Comment:	Expect delay from selecting COM port until ready so keep open for fastest Updates.

Property:	**DIFcount**
Description:	Returns count of negative transitions on DIF input.
Example:	Counter = FTview1.DIFcount
Comment:	Totalizing counter cannot be written (prevents missing counts).

Property:	**DIFstate**
Description:	Returns logic level of DIF input pin at time of Update.
Example:	DIFinput = FTview1.DIFstate
Comment:	Digital input (0 or 1) when DIF not used as counter. DIN and DIS operate identically.

Property:	**DITinterval**
Description:	Returns time between negative transitions on DIT input.
Example:	Period = FTview1.DITinterval
Comment:	Invert interval for frequency.

Property:	**DITstate**
Description:	Returns logic level of DIT input at time of Update.
Example:	DITinput = FTview1.DITstate
Comment:	Digital input (0 or 1) when DIT not used as timer. DIN and DIS operate identically.

Property:	**DOEstate**
Description:	Sets logic level of DOE output pin.
Example:	FTview1.DOEstate = 1
Comment:	Digital output (0 or 1). DON, DOK, and DOS operate identically.

Property:	**DOGduty**
Description:	Sets percent duty cycle of DOG pulse width modulated output.
Example:	FTview1.DOGduty = 25
Comment:	Modulate average output levels without frequent adjustment.

Property: **DPWduty**
Description: Sets percent duty cycle of DPW pulse width modulated output.
Example: FTview1. DPW duty = 50
Comment: Modulate average output levels without frequent adjustment.

Property: **FWver**
Description: Returns firmware version of micro. "NA" means no micro responded on Update.
Example: Firmware = FTview1.FWver
Comment: Distinguish between future revisions.

Property: **IDstring**
Description: Unique identifier label saved in micro nonvolatile EEPROM.
Example: FTview1.IDstring = "Flextek"
Comment: Distinguish between multiple micros on bus.

Property: **MicroAddress**
Description: Address of micro on multi-drop serial data bus.
Example: FTview1.MicroAddress = 0
Comment: Communicate with multiple micros from single serial port.

Property: **MicroStatus**
Description: Returns "OK" if valid data or "Busy" if checking or "None" if no micro found.
Example: MicStat = FTview1.MicroStatus
Comment: Determine if micro available for communication. UpdateCmd ignored if "Busy"

Event: **NewData**
Description: Event fired when new data is ready after completion of UpdateCmd (data write and read).
Example: Private Sub FTview1_NewData()
Comment: Sub executes tasks on new data. Update within sub loops at update rate until ClearCmd.

Property: **PWMduty**
Description: Sets percent duty cycle of PWM pulse width modulated output.
Example: FTview1. PWM duty = 75
Comment: Modulate average output levels without frequent adjustment.

Property: **RSTstate**
Description: Reset status latched high if RST pin low or 5VO brownout. Clear RST in software.
Example: FTview1.RSTstate = 0
Comment: Useful flag to detect power or noise problems. RST cannot be set by software

Property: **UpdateCmd**
Description: Command to write new values to micro and read all registers.
Example: FTview1.UpdateCmd = True
Comment: Micro communication only on Update command. Generates NewData event when done.

Property: **Vrefvolt**
Description: Reference voltage correction factor saved permanently in micro EEPROM.
Example: FTview1.Vrefvolt = 5.002
Comment: Measure 5VO and save in micro to automatically calibrate ADC measurements.

Property: **ZeroCmd**
Description: Sets all outputs to zero so pins low on next Update.
Example: FTview1.ZeroCmd = True
Comment: Provides fast shutdown of output power drivers.