Digitally Controlled Synchronous Buck Converter

Taufik[1*], M. Starman[1], M. Anwari[2]

[1]*Cal Poly State University, San Luis Obispo, CA 93407, USA*
[2]*Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia*

The demand for high performance, more flexible and reconfigurable dc-dc converters has pushed dc-dc converter designers to seek for alternative ways to control the pulse-width modulated switch inside the dc-dc converter. During the past few years, efforts have been put forward both by academics and power semiconductor industries to phase out the most commonly used analog PWM controllers with digital controller with its associated software. This paper presents an experimental result of designing and building a 5W synchronous buck converter that employs a microcontroller to process the output voltage signal and translate it into PWM signal to control the switches inside the buck converter and hence regulating the output voltage. Design requirements, procedures, component selections and measurement results will be described in this paper. In addition, operating performance of the buck converter as a result of implementing the digital controller will also be discussed, along with the description of the actual schematics and the final printer circuit board of the circuit.

1. Introduction

Recent advancement in economical but yet powerful in the marketplace has caused digital solutions to enter a number of applications traditionally known to be purely analog systems. A great example of this would be in the area of motor controls. Modern high-performance motor controllers are expected to concurrently achieve several objectives. They must meet or exceed the dynamical specifications, and other requirements such as high efficiency, power factor, and electromagnetic compatibility (EMC). To meet all these requirements, designers are turning to mathematical algorithms that are computationally demanding. This is the domain where digital control will suit perfectly. In fact, most if not all high-quality electric motors today are digitally controlled implementing Digital Signal Processors (DSPs) or microcontrollers where motor speed and torque can be varied and precisely set.

Like any other technological methods, digital control also has pros and cons. Digital controllers offer several benefits as summarized below [(1)]:

- Provision of new capabilities such as implementation of advanced algorithms enabling higher performance, and lower energy consumption, among other things.
- Immunity to drifts since digital controller's functioning is substantially unaffected by either time or temperature drifts. Equations in software do not drift, unlike analog controllers.
- Software implemented on programmable controllers can calibrate out the inaccuracies and can automate this calibration process, hence lowering the cost of manufacturing by eliminating a manual calibration step.
- Ease of implementation since functions are easily implemented in software.
- Faster time to market since digital controllers make it possible to leverage existing off-the-shelf controllers, which allow the fastest realization of a design. In addition, the design of controllers is often an iterative process, with repeated design and test steps, until the specifications are met. Such an iterative process can be executed rapidly by means of a software-configurable controller.
- Control law changes are done by software updates, hence a much faster process than incorporating these changes with hardware.
- Far less sensitive to component tolerances since software in digital controllers are far less susceptible to component tolerances.

The arrival of digitally controlled power supplies, though, raises some significant performance issues. Analogue controllers are based on continuous time, so performance usually is not a limiting factor. Digital controllers, on the other hand, must address time quantization effects. Such controllers are driven by a system clock that generates granular time steps. The steps depend on the system clock frequency and the switching frequency. The resolution (in bits) of a digital pwm controller is proportional to system clock speed ($f_{system\text{-}clock}$) and is inversely proportional to the switching frequency (f_{PWM}), as indicated by (2):

$$\text{Resolution} = \log_2 (f_{system\text{-}clock}/f_{PWM}) \quad (1)$$

Table 1 summarizes the resolution of various combinations of system clock and PWM frequencies [(2)]. Table 1 also shows that with higher PWM frequencies, the resolution can rise to unacceptable levels.

Table 1. Processor clock frequency

PWM	100			150		
(kHz)	Bits	%	Instructions	Bits	%	Instructions
20	12.3	0.02	5000	12.9	0.01	7500
60	10.7	0.06	1667	11.3	0.04	2500
120	9.7	0.12	833	10.3	0.08	1250
250	8.6	0.25	400	9.2	0.17	600
300	8.4	0.30	333	9.0	0.20	500
500	7.6	0.50	200	8.2	0.33	300
750	7.1	0.75	133	7.6	0.50	200
1000	6.6	1.00	100	7.2	0.67	150
2000	5.6	2.00	50	6.2	1.33	75
4000	4.6	4.00	25	5.2	2.67	38

* Taufik. E-mail: taufik@calpoly.edu

ISBN 978-979-16338-0-2

2. Requirements, Design, and Results

The buck converter presented in this paper is a synchronous 5V 5W buck converter whose input voltage may range from 10V to 14V with a nominal value of 12V. Figure 1 shows the power stage of a basic buck converter. In synchronous topology, however, the diode is replaced with another switch, as was done for this project.

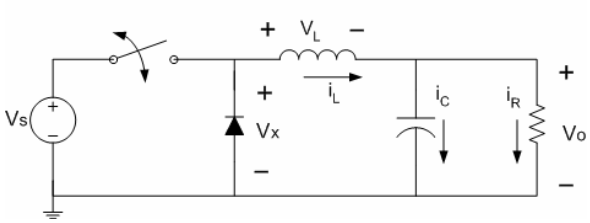

Fig. 1. Buck converter power stage

Based on continuous conduction mode operation of the converter, the following main component values were obtained:

- Switch

Power MOSFET 60V 9.9A DPAK. With these ratings, the same power MOSFET was being used in both the main switch and the synchronous switch.

- Inductor

The inductance value chosen was 115uH. This value was obtained by having two 30uH inductors in parallel. The parallel connection should give the benefit of decreased copper resistance while splitting the output current which in turn helps in efficiency.

- Capacitor

The value selected was 473uF which was achieved by placing three capacitors in parallel. Doing so will effectively reduce the equivalent series resistance (ESR) hence helps with the overall efficiency while obtaining the desired capacitance value.

- Digital Controller

Microchip's PIC12F683 microcontroller was chosen for this design. It is a low pin-count PIC® flash microcontroller (figure 2) with standardized features including a wide operating voltage of 2.0-5.5 volts, on-board EEPROM Data Memory, and nanoWatt Technology. Standard analog peripherals include up to 4 channels of 10-bit A/D, an analog comparator module with a single comparator, programmable on-chip voltage reference and a Standard Capture and Compare and PWM (CCP) module. Summary of its functionalities is given in table 2 [(3)]. A program was written for this PIC as shown in detail in appendix A with PWM frequency set at 208.3 kHz.

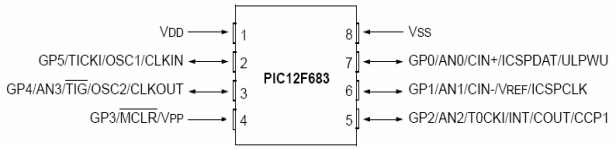

Fig. 2. PIC12F683 pinouts

Table 2. PIC12F683 features

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)				
PIC12F683	2048	128	256	6	4	1	2/1

The final circuit schematic for the buck converter is shown in figure 3. The MOSFET driver was used to process the output of the microcontroller to drive the two MOSFETs of the buck at the appropriate gate levels. The front-end linear regulator was needed to tap in the input voltage to power the microcontroller. This way one dc power source was needed for both the buck and the PIC microcontroller. A type 2 error amplifier as indicated by the series-parallel RC at the output voltage sensing was adopted and was enough to maintain a stable operation of the buck converter.

The final version of the circuit was built on a PCB as depicted in figure 4. The PCB was being used to reduce possible switching noises and interferences caused by the PWM signals. Due to time constraints, the size of the board was not optimized. Component selections could also be improved should there be no cost limitation. There are several ways to reduce the size of the board. First, the main inductors are noticeably not optimized since two relatively big inductors are employed. This could be simplified by using a single inductor with bifiliar configuration which is basically two inductors in parallel connection but wound on a single

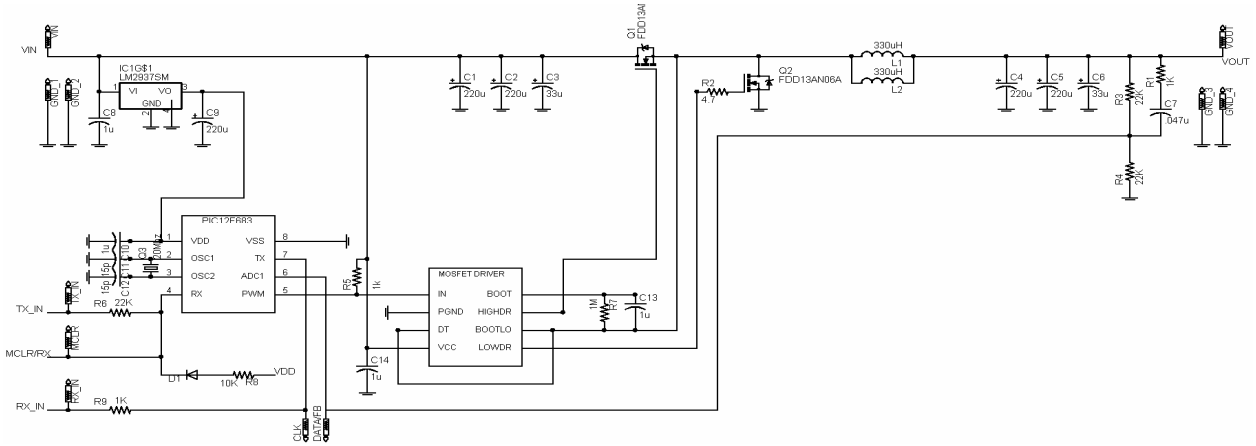

Fig. 3. Final circuit schematic

magnetic core. The size of the board may also be significantly reduced by more careful component selections such as the capacitors and the MOSFETs. With the advent of semiconductor packaging technology, there are currently available power MOSFETs with the same ratings as the ones used in this project but yet come in a smaller package. Replacing capacitor cans with say ceramic surface mount capacitors will be another means of reducing the size of the board. Furthermore, adding a schottky diode across the synchronous MOSFET to provide the freewheeling current during dead time may also slightly increase the overall efficiency. The redesigning of the board to double-sided design will also reduce the overall size of the PCB.

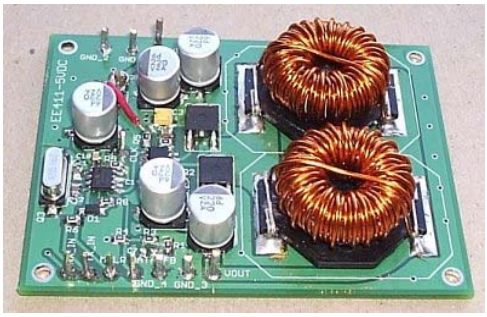

Fig. 4. Final printed circuit board (PCB)

The digitally controlled buck converter was tested and was found to be fully operational. The line regulation was measured to be less than 2% when the input was changed from 10V to 14V while the output current was held at the maximum value of 3 Amps. The load regulation was found to be less than 1% and it was measured between 10% load to 90% load while the input voltage was held at its nominal value of 12V. The output voltage peak to peak ripple was also measured at full load and it was found to be extremely good of less than 0.5%. Lastly, the overall efficiency measurement of the converter was conducted. The result is illustrated in figure 5. The figure indicates that the peak efficiency is at 90% between about 25% to 35% load, while at full load the efficiency is observed to be at 86%.

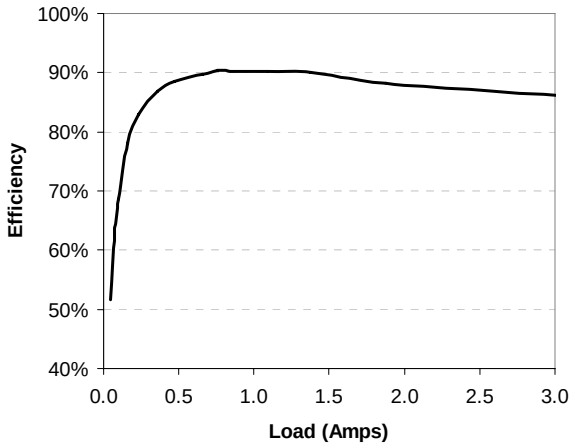

Fig. 5. Efficiency plot of the digitally controlled buck

5. Conclusion

In this paper, a digitally controlled buck converter has been presented. Its design including main component selections was discussed. One benefit that might be obtained by the use of digital control is the overall size of the board. Unfortunately, due to time and cost constraints, the resulting buck converter in this project does not really point this out. Therefore, issues dealing with the pcb board size were discussed along with possible methods to improve the board size. Results of measurements done on the buck converter were presented and showed very promising capability of digital control used in dc-dc converters. Even without component and board optimizations, the efficiency of the buck at full load was already high at 86%, while the peak efficiency was at 90%. With more time and more elaborate design, the efficiency could be boosted up to around 95%. Future improvement of this buck converter will address the component sizing, layout, and board design. Although the programming aspect was not described in the paper, the detail of source code used in the microcontroller is provided in the appendix to follow as a reference for readers.

References

(1) K. Godbole, “Converting Analog Controllers to Smart Controllers with the TMS320C2000 DSPs,” Application Report, Texas Instruments, June 2004.
(2) S. Chon, “Overcoming The Challenges Of Moving To Full Digital Power-Supply Control,” *Electronic Design Magazine,* August 2006.
(3) http://ww1.microchip.com/downloads/en/DeviceDoc/41211D_.pdf, PIC12F683 datasheet from Microchip website.

Appendix A. Source Code

```
'PIN SETTINGS (0 OUTPUT, 1 INPUT)(BIT76543210)
TRISIO = %00111010   '-|-|OSC1|OSC2|RX|PWM|ADC|TX
'REGISTER SETTINGS
OSCCON = %00000000  'SET OSCILLATOR TO EXTERNAL
INTCON = %11001000  'ENABLE GLOBAL AND EXTERNAL INTERRUPTS
IOC = %00001000     'INTERRUPT ON SERIAL INPUT
ADCON0 = %10000110  'ADC RIGHT JUSTIFY, VREF=VDD, AQUISITION AND CONVERSION
TIME        ??
CCP1CON =   %00001100
PR2 = %00010111     'SET PWM FREQUENCY TO 208.3khz
CCPR1L = %00000011  'MSB FOR PWM DUTY CYCLE  BITS 9,8,7,6,5,4,3,2
CCP1CON.5 = 1       'LSB FOR PWM DUTY CYCLE  BIT 1
CCP1CON.4 = 1       'LSB FOR PWM DUTY CYCLE  BIT 0
T2CON = %00000100   'ENABLE TIMER 2, SET PRESCALER
'ALIASES
PWMOUT var GPIO.2     'PWM OUTPUT FOR MOSFET DRIVE
'PICBASIC DEFINES
DEFINE OSC 20              'SET CLOCK FREQUENCY TO 48MHZ
DEFINE ADC_BITS 10         '10-BIT ADC
DEFINE ADC_SAMPLEUS 1
define DEBUG_REG GPIO       'DEBUG ON PORTC
DEFINE DEBUG_BIT 0          'DEBUG PIN 6 OF PORT C
DEFINE DEBUG_BAUD 9600      'DEBUG BAUDRATE 57600KBS
DEFINE DEBUG_MODE 1         'DEBUG MODE SET TO INVERTED
DEFINE DEBUGIN_REG GPIO     'Debugin pin port
DEFINE DEBUGIN_BIT 3        'Debugin pin bit
DEFINE DEBUGIN_MODE 1       'Debugin mode: 0 = True, 1 = Inverted
'VARIABLES
_DUTY_CYCLE VAR BYTE
_OUTPUT_VOLTAGE_OLD3 VAR WORD
_OUTPUT_VOLTAGE_OLD2 VAR WORD
_OUTPUT_VOLTAGE_OLD1 VAR WORD
_OUTPUT_VOLTAGE VAR WORD
_OUTPUT_VOLTAGE_SAMPLE var word
_VB_COMMAND VAR WORD
_VPARAM VAR WORD
_FPARAM VAR WORD
i var word
j var word
RAISEFLAG VAR WORD
'INITILIZATION
CLEAR
PAUSE 50
DEBUG "INIT",10
Read 0,_VPARAM.Byte0
```

```
read 1,_VPARAM.Byte1
read 2,_FPARAM.Byte0
read 3,_FPARAM.Byte1
   select CASE _FPARAM
         case 0
            PR2 = $17
            T2CON = %00000100
         case 1
            PR2 = $1F
            T2CON = %00000100
         case 2
            PR2 = $3F
            T2CON = %00000100
         case 3
            PR2 = $FF
            T2CON = %00000100
         case 4
            PR2 = $FF
            T2CON = %00000101
         case 5
            PR2 = $FF
            T2CON = %00000111
   end select
ON INTERRUPT GOTO SERIAL_INTERRUPT
'INITIAL DUTY:
   'LOAD FROM EPPROM
   'SET DUTY CYCLE
'    _DUTY_CYCLE = 44
 '       CCPR1L = (_DUTY_CYCLE/2/2)
  '      CCP1CON.5 = _DUTY_CYCLE.1
   '     CCP1CON.4 = _DUTY_CYCLE.0
MAIN:
    GOSUB CHECKOUTPUT
   'DEBUG dec _OUTPUT_VOLTAGE,10
     _DUTY_CYCLE = (CCPR1L*2*2)+(CCP1CON.5*2)+(CCP1CON.4)
        IF _OUTPUT_VOLTAGE > _VPARAM THEN
           IF _DUTY_CYCLE != 0 THEN
              'DEBUG "-",10
              _DUTY_CYCLE = _DUTY_CYCLE - 1
              CCPR1L = (_DUTY_CYCLE/2/2)
              CCP1CON.5 = _DUTY_CYCLE.1
              CCP1CON.4 = _DUTY_CYCLE.0
           ENDIF
        endif
        if _OUTPUT_VOLTAGE < _VPARAM THEN
           IF _DUTY_CYCLE != 255 THEN
              'DEBUG "+",10
              _DUTY_CYCLE = _DUTY_CYCLE + 1
              CCPR1L = (_DUTY_CYCLE/2/2)
              CCP1CON.5 = _DUTY_CYCLE.1
               CCP1CON.4 = _DUTY_CYCLE.0
           ENDIF
        else
           GOTO STEADYSTATE
        ENDIF
    GOTO MAIN
STEADYSTATE:
   'DEBUG "SS",10
   GOSUB CHECKOUTPUT
   GOSUB CHECKOUTPUT
   GOSUB CHECKOUTPUT
   IF ABS(_OUTPUT_VOLTAGE_OLD3 -_VPARAM) > 10 THEN
        DEBUG #ABS(_OUTPUT_VOLTAGE_OLD3 -_VPARAM),10
        IF ABS(_OUTPUT_VOLTAGE_OLD2 -_VPARAM) > 10 THEN
             DEBUG #ABS(_OUTPUT_VOLTAGE_OLD2 -_VPARAM),10
           IF ABS(_OUTPUT_VOLTAGE_OLD1 -_VPARAM) > 10 THEN
                DEBUG #ABS(_OUTPUT_VOLTAGE_OLD1 -_VPARAM),10,10
                GOTO MAIN
           ENDIF
        ENDIF
   ENDIF
   GOTO STEADYSTATE
CHECKOUTPUT:
   _OUTPUT_VOLTAGE_OLD3 = _OUTPUT_VOLTAGE_OLD2
   _OUTPUT_VOLTAGE_OLD2 = _OUTPUT_VOLTAGE_OLD1
   _OUTPUT_VOLTAGE_OLD1 = _OUTPUT_VOLTAGE
   for i = 1 to 32
      ADCIN 1, _OUTPUT_VOLTAGE_SAMPLE
      _OUTPUT_VOLTAGE = _OUTPUT_VOLTAGE + _OUTPUT_VOLTAGE_SAMPLE
   next i
   _OUTPUT_VOLTAGE = (_OUTPUT_VOLTAGE / 32)
   RETURN
disable interrupt
SERIAL_INTERRUPT:
INTCON.0 = 0
      DEBUGIN 5,EXITINT,[_VB_COMMAND]
      SELECT CASE _VB_COMMAND
      case "V"
         DEBUGIN 5,EXITINT,[DEC3 _VB_COMMAND]
         'DEBUG "V=",dec _VB_COMMAND,10
         _VPARAM = _VB_COMMAND
         WRITE 0,_VPARAM.Byte0
         WRITE 1,_VPARAM.Byte1
         DEBUGIN 5,EXITINT,[_VB_COMMAND]
      CASE "F"
         DEBUGIN 5,EXITINT,[DEC3 _VB_COMMAND]
         'DEBUG "F=",dec _VB_COMMAND,10
         _FPARAM = _VB_COMMAND
         WRITE 2,_FPARAM.Byte0
         WRITE 3,_FPARAM.Byte1
         select case _FPARAM
         case 0
            PR2 = $17
            T2CON = %00000100
         case 1
            PR2 = $1F
            T2CON = %00000100
         case 2
            PR2 = $3F
            T2CON = %00000100
         case 3
            PR2 = $FF
            T2CON = %00000100
         case 4
            PR2 = $FF
            T2CON = %00000101
         case 5
            PR2 = $FF
            T2CON = %00000111
         end select
      end select
      INTCON.0 = 0
EXITINT:
   resume
enable interrupt
end
```

ISBN 978-979-16338-0-2