



Praktikum Mikrocontroller und DSP (EMD)

Prof.Dr.-Ing. W. Caninenberg

PWM-Motorsteuerung am XC167 - Mikrocontroller

Name, Vorname	Matrikelnummer
Versuchstag	Testat

1. Aufgabenstellung

In diesem Versuch soll mit einem Infineon xc167ci Mikrocontroller ein PWM-Signal (z.B. zur Motorsteuerung) erzeugt werden.

Bei der Pulsweitenmodulation (PWM) wird ein Rechtecksignal mit variabler Impulsbreite (Tastverhältnis) und variabler Frequenz erzeugt.

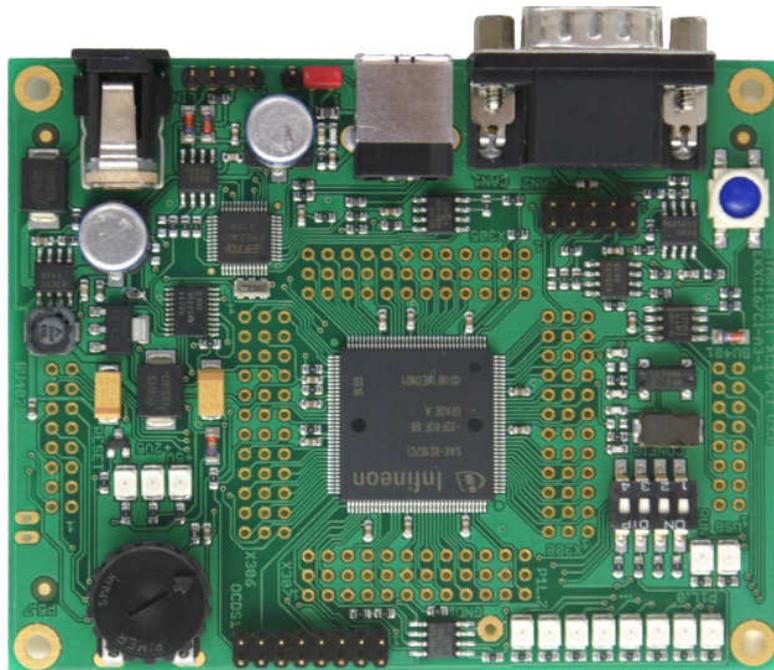
Die zur Berechnung der PWM-Parameter für Ihren Versuch erforderlichen Werte für Frequenz und Tastverhältnis erhalten Sie vor Versuchsbeginn.

2. Versuchsvorbereitung

- Suchen Sie die Unterlagen für den Infineon xc167 im Internet. Dabei ist insbesondere das User Manual V2.0 Volume 2 (Peripheral Units) interessant.
- Lesen Sie den Abschnitt 18 (Capture/Compare Unit CAPCOM6) aus dem User-Manual des xc167.
- Versuchen Sie anhand des Datenblatts und dieser Anleitung die zur Programmierung der CAPCOM6-Einheit erforderlichen Register zu verstehen. Sie müssen diese Register im Praktikum berechnen!

3. Aufbau der Versuchsanordnung

Die Versuchsanordnung besteht aus einem xc167-Evaluation-Board der Firma Infineon.



Entwicklungsboard

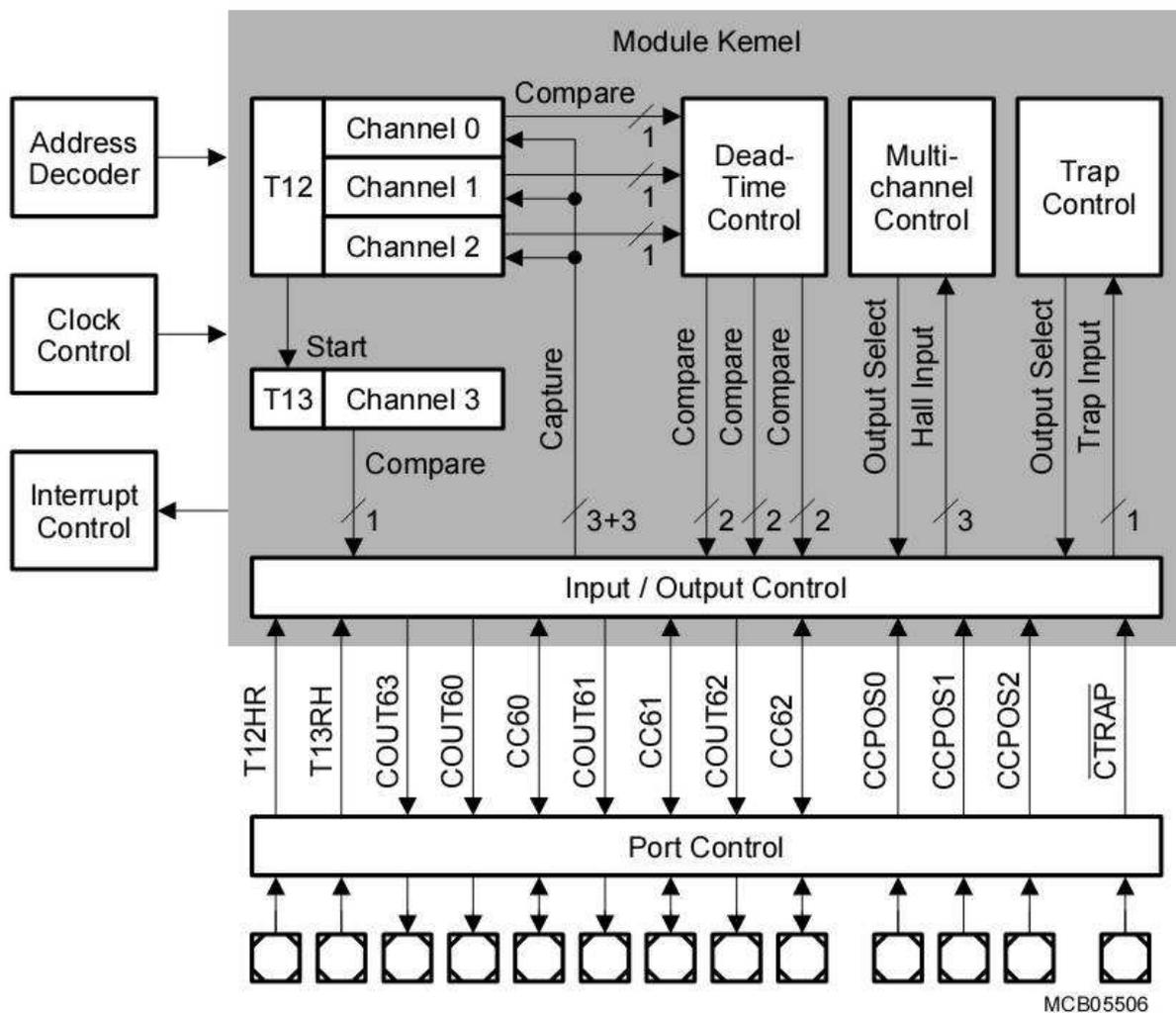
Zusätzlich zu diesem Entwicklungsboard finden Sie an jedem Versuchsaufbau ein Messgerät mit einer Frequenzanzeige oder eine Zusatzplatine mit einem Motor und der dazu gehörenden Ansteuerschaltung. Ein auf der Motorachse befindlicher Tachogenerator erzeugt eine von der Motordrehzahl abhängige Spannung. Diese Spannung wird auf einem DVM angezeigt.

Der zur Softwareentwicklung eingesetzte Compiler ist die Evaluation-Version des Keil C167-Compilers (uVision V3).

4. C167 CAPCOM6

Der xc167ci hat, im Gegensatz zum einigen anderen c167-Versionen, keine eigene PWM-Einheit. PWM-Signale müssen daher mit der CAPCOM (Capture / Compare) Einheit des xc167 erzeugt werden.

Der xc167 ist mit zwei Capture/Compare-Einheiten ausgestattet (CAPCOM2 und CAPCOM6), die eine Vielzahl von Optionen zur Erzeugung und Messung von Impulsen bieten. Wir verwenden hier CAPCOM6.



Die Capture-Betriebsarten der CAPCOM-Einheit dienen zur Messung und Auswertung von externen Signalen (wird hier nicht benötigt). Die Compare-Betriebsarten dienen zur Erzeugung von PWM-Signalen. Die drei 16-Bit Kanäle von CAPCOM6 werden von Timer T12 gesteuert und können jeweils zwei Ausgänge ansteuern. Ein weiterer 10-Bit Compare-Kanal wird von Timer T13 angesteuert.

Prinzipiell werden hier zwei Betriebsarten unterstützt:

Edge Aligned Mode:

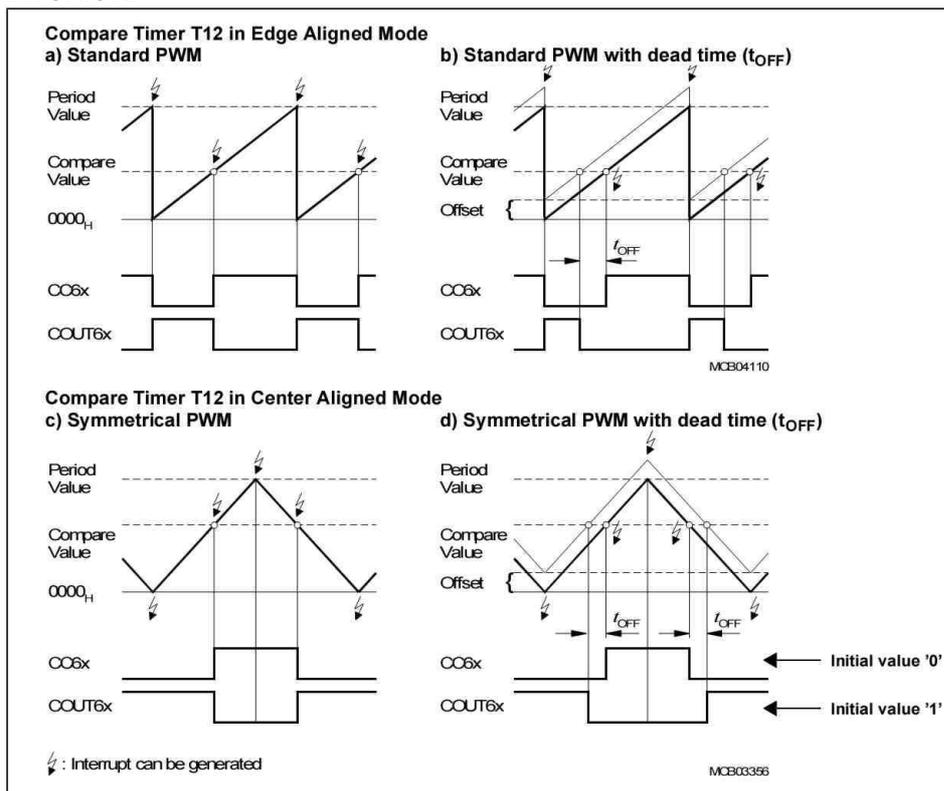
Der Compare-Timer fängt mit dem Startwert 0x0000 an zu Zählen. Beim Erreichen des Periodenwertes in **CCU6_12PR** wird der Timer gelöscht und neu gestartet. Beim Rücksetzen des Timers werden außerdem die Ausgänge in den Ruhezustand zurückgeschaltet.

Wenn der Timer den Compare-Wert **CCU6_CC6xR** (beschrieben über **CCU6_CC6xSR**) erreicht wird der Ausgangswert invertiert.

Center Aligned Mode:

Der Compare-Timer fängt mit dem Startwert 0x0000 an zu Zählen. Beim Erreichen des Periodenwertes in **CCU6_T12PR** wird die Zählrichtung umgekehrt und der Zähler zählt abwärts bis 0.

Auch hier wird beim Erreichen des Compare-Werts **CCU6_CC6xR** der Ausgangswert invertiert.



Der in beiden Operationsarten verwendete Timer T12 wird dabei mit einer Frequenz f_{T12} getaktet die über das Register **CCU6_TCTR0** von f_{CC6} bis $f_{CC6}/32768$ eingestellt werden kann.

Die von der CAPCOM6-Einheit erzeugten Signale sind charakterisiert durch die Dauer ihrer aktiven und passiven Phasen. Daraus ergeben sich die Signalfrequenz und das Tastverhältnis des erzeugten Signals. Um diese Ausgangssignale an die Erfordernisse der an den Ausgängen angeschlossenen Schaltung anzupassen kann der Logikpegel für den aktiven bzw. passiven Zustand durch das Register **CCU6_PSLR** eingestellt werden.

Die erzeugten Signale können an den Pins **CC6x** bzw. **COUT6x** ausgegeben werden (Pins P1L.0-P1L.5). Die dazu erforderliche Ausgangskonfiguration erfolgt über mehrere Register.

5. Erforderliche Register

Aus der Sicht des Programmierers besteht die CAPCOM6-Einheit des xc167 aus einer Reihe von SFRs (Special Function Registers)

Data Registers		Control Registers		Interrupt Control		System Registers	
T12	X	CMPSTAT	X	CCU6_IS	X	P1L	
T12PR	X	CMPMODIF	X	CCU6_ISS	X	DP1L	E
T12DTC	X	TCTR0	X	CCU6_ISR	X	ALTSEL0P1L	E
CC60R	X	TCTR2	X	CCU6_INP	X		
CC60SR	X	TCTR4	X	CCU6_IEN	X	P1H	
CC61R	X	MODCTR	X			DP1H	E
CC61SR	X	TRPCTR	X	CCU6_T12IC	E	ALTSEL0P1H	E
CC62R	X	PSLR	X	CCU6_T13IC	E		
CC62SR	X	MCMOUTS	X	CCU6_EIC	E	SYSCON3	E
T13	X	MCMOUT	X	CCU6_IC	E		
T13PR	X	MCMCTR	X				
CC63R	X	T12MSEL	X				
CC63SR	X						

T12/T13	CAPCOM6 Timer T12/T13 Register	TRPCTR	Trap Control Register
T12PR/T13PR	CAPCOM6 T12/T13 Period Register	PSLR	Passive State Level Register
T12DTC	CAPCOM6 T12 Deadtime Control Reg.	IS	Interrupt Status Register
TCTR _x	CAPCOM6 Timer Control Register	ISS/ISR	Interrupt Status Set/Reset Register
T12MSEL	T12 Capture/Compare Mode Select Reg.	IEN	Interrupt Enable Register
CC6 _x R	CAPCOM6 Capture/Compare Register	INP	Interrupt Node Pointer Register
CC6 _x SR	CAPCOM6 Capt./Comp. Shadow Reg.	T12IC/T13IC	CAPCOM6 Timer Interrupt Control Reg.
CMPSTAT	Capture/Compare State Register	CCU6_IC	CAPCOM6 Interrupt Control Register
CMPMODIF	Capture/Comp. State Modification Reg.	CCU6_EIC	CAPCOM6 Error Interrupt Control Reg.
MODCTR	Modulation Control Register	P1L/P1H	PORT1 Data Register
MCMOUT	Multi-Channel Mode Output Register	DP1L/DP1H	PORT1 Direction Control Register
MCMOUTS	Multi-Ch. Mode Output Shadow Reg.	ALTSEL0P1 _x	PORT1 Alternate Outp. Select Reg. 0
MCMCTR	Multi-Channel Mode Control Register	SYSCON3	System Control Reg. 3 (Per. Mgmt.)

mc_capcom60100_registers.vsd

Timer Control Register 0

CCU6_TCTR0

Timer Control Register 0

XSFR (E8AC_H/--)

Reset Value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	STE 13	T13R	T13 PRE	T13CLK		CTM	CDIR	STE 12	T12R	T12 PRE	T12CLK			
-	-	rh	rh	rw	rw		rw	rh	rh	rh	rw	rw			

Field	Bits	Type	Description
CTM	7	rw	T12 Operating Mode 0 Edge-Aligned Mode 1 Center-Aligned Mode
T12PRE	3	rw	Timer T12 Prescaler Enable Bit Enables the additional 1/256-prescaler of T12 0 The additional prescaler is disabled 1 The additional prescaler is enabled
T12CLK	[2:0]	rw	Timer T12 Input Clock Select Selects the input clock for Timer T12 which is derived from the CAPCOM6 input clock according to the equation: $f_{T12} = f_{CC6} / 2^{<T12CLK>}$. See Table 18-1 .

Aus diesem Register werden für unseren Anwendungsfall die Bits **CTM**, **T12PRE** und **T12CLK** benötigt. Alle weiteren Bits können in ihrem RESET-Zustand belassen werden. Mit dem Bit **CTM** wird die PWM-Betriebsart ausgewählt. Mit **T12PRE** und **T12CLK** wird der Vorteiler für T12 eingestellt.

T12CLK	Resulting Input Clock Prescaler Off (T12PRE = 0)	Resulting Input Clock Prescaler On (T12PRE = 1)
000 _B	f_{CC6}	$f_{CC6}/256$
001 _B	$f_{CC6}/2$	$f_{CC6}/512$
010 _B	$f_{CC6}/4$	$f_{CC6}/1024$
011 _B	$f_{CC6}/8$	$f_{CC6}/2048$
100 _B	$f_{CC6}/16$	$f_{CC6}/4096$
101 _B	$f_{CC6}/32$	$f_{CC6}/8192$
110 _B	$f_{CC6}/64$	$f_{CC6}/16384$
111 _B	$f_{CC6}/128$	$f_{CC6}/32768$

Timer Control Register 4

CCU6_TCTR4

Timer Control Register 4

XSFR (E8A6_H/--)

Reset Value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13 STD	T13 STR	-	-	-	T13 RES	T13 RS	T13 RR	T12 STD	T12 STR	-	-	DT RES	T12 RES	T12 RS	T12 RR
W	W	-	-	-	W	W	W	W	W	-	-	W	W	W	W

Field	Bits	Type	Description
T13STR T12STR	14 6	w	Timer T13/T12 Shadow Transfer Request 0 No action 1 STE13/STE12 is set, requesting the shadow transfer
T13RS T12RS	9 1	w	Timer T13/T12 Run Bit Set Control¹⁾ Software can set bit T13R/T12R (start timer T13/T12) by writing to bit T13RS/T12RS. 0 T13R/T12R is not set 1 T13R/T12R is set, T13/T12 starts counting

Das Timer-Control-Register CCU6_TCTR4 enthält nur zwei Bits die uns interessieren. Einige der Register der CAPCOM6-Einheit können nicht direkt beschrieben werden. Um Daten in diese Register zu übertragen wird zunächst ein sogenanntes Shadow-Register beschrieben. Die Inhalte aller Shadow-Register können gleichzeitig durch das Setzen des Bits **T12STR** in die eigentlichen Register übertragen werden.

Das Setzen des Bits **T12RS** startet den Timer.

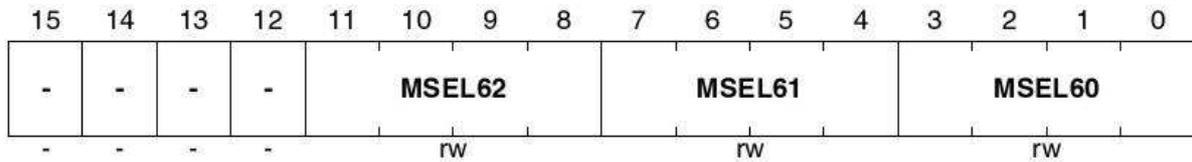
T12 Mode Select Register

CCU6_T12MSEL

T12 Mode Select Register

XSFR (E8C6_H/--)

Reset Value: 0000_H



Field	Bits	Type	Description
MSEL62	[11:8]	rw	Capture/Compare Mode Selection These bitfields select the operating mode of the three T12 capture/compare channels. Each channel (x = 0, 1, 2) can be programmed individually for one of these modes (except for Hall Sensor Mode). See Table 18-2 .
MSEL61	[7:4]		
MSEL60	[3:0]		

Table 18-2 Capture/Compare Modes Overview

MSEL6 _x	Selected Operating Mode
0000 _B	Compare outputs disabled, pins CC6 _x and COUT6 _x can be used for IO.
0001 _B	Compare output on pin CC6 _x , pin COUT6 _x can be used for IO.
0010 _B	Compare output on pin COUT6 _x , pin CC6 _x can be used for IO.
0011 _B	Compare output on pins COUT6 _x and CC6 _x .
01XX _B	Double-Register Capture modes, see Chapter 18.1.4 and Table 18-3 .
1000 _B	Hall Sensor Mode, see Chapter 18.5 . In order to properly enable this mode, all three MSEL6 _x fields have to be programmed to Hall Sensor mode.
1001 _B	Hysteresis-like mode, see Chapter 18.1.5 .
101X _B 11XX _B	Multi-Input Capture modes, see Chapter 18.1.4 and Table 18-4 .

Das Mode-Select-Register wird benötigt um die Betriebsart für die sechs Ausgänge der PWM-Einheit festzulegen. Für jeden für PWM benötigten Ausgang muss das entsprechende Bit in diesem Register als Compare-Output konfiguriert werden.

Passive State Level Register

CCU6_PSLR

Passive State Level Register

XSFR (E8C4_H/--)

Reset Value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	PSL 63	-	PSL					
-	-	-	-	-	-	-	-	rwh	-	rwh					

Field	Bits	Type	Description
PSL63	7	rwh	T13 Output COUT63 Passive State Level Control This bitfield defines the passive level of the output pin COUT63. 0 The passive level is 0 1 The passive level is 1
PSL	[5:0]	rwh	T12 Outputs Passive State Level Control Defines the passive level driven by the module outputs during the passive state. 0 The passive level is 0 1 The passive level is 1 PSL[5:0] corresponds to (left to right): COUT62, CC62, COUT61, CC61, COUT60, CC60.

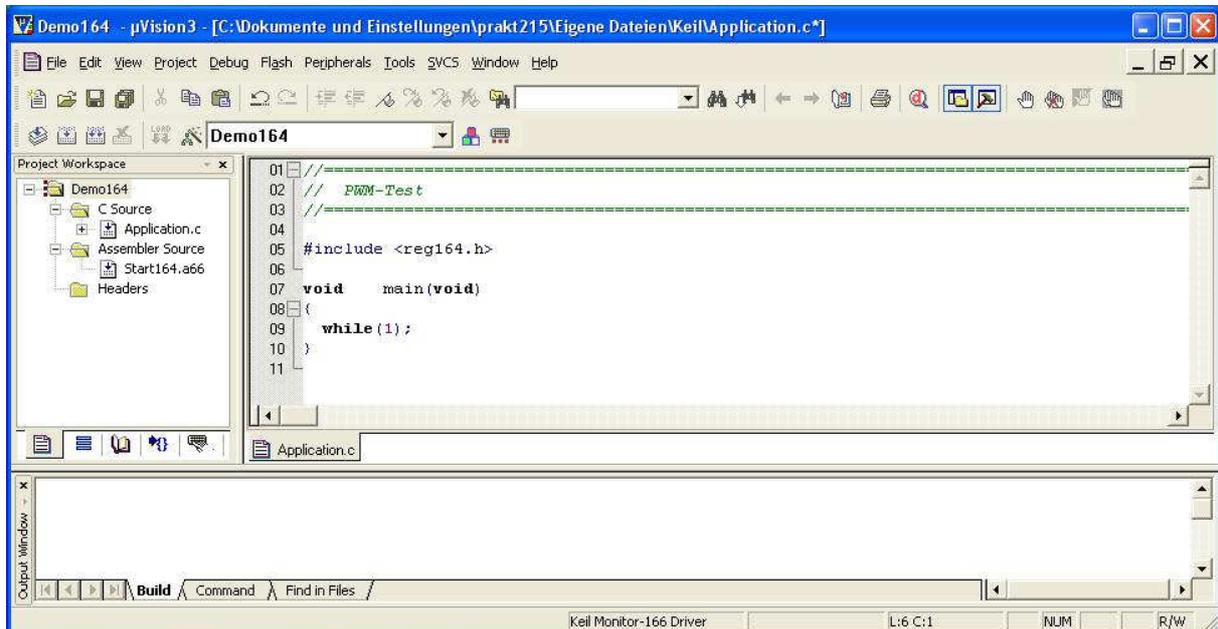
Das Passive State Level Register ermöglicht mit einem Bit für jeden der sechs Ausgänge das Verhalten dieses Ausgangs zu invertieren.

Neben diesen Registern werden die bereits besprochenen Register **CCU6_T12PR** und die Register **CCU6_CC60SR-CCU6_CC62SR** benötigt.

Weiterhin ist es erforderlich die Pinconfiguration der benötigten Ausgangspins über die Register **ALTSEL0P1L** und **DP1L** einzustellen.

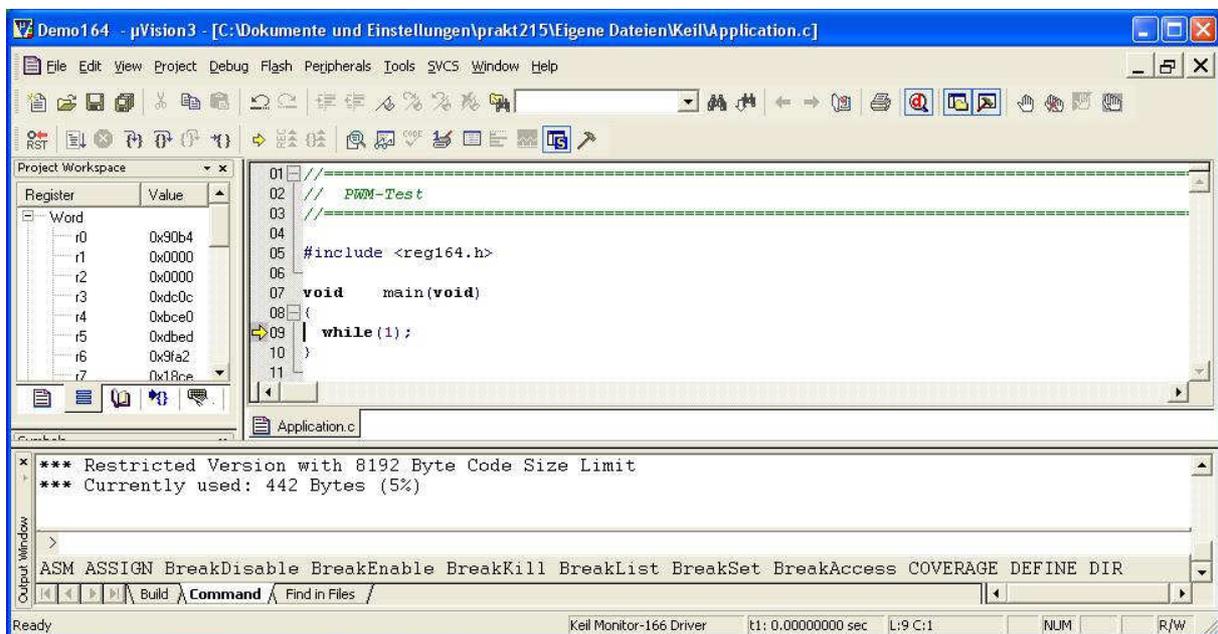
6. Der Keil C166-Compiler

Nach dem Start der Keil-Entwicklungsumgebung μ Vision3 sehen Sie folgende Oberfläche:



Nach Eingeben des Quelltextes starten Sie den Compiler durch Klicken auf 

Wenn Ihr Quelltext ohne Fehler übersetzt wird können Sie anschließend mit  den Debugger starten. Sie bekommen dann folgendes Fenster angezeigt:



Ein Klick auf  führt Ihr Programm aus.

Sie haben dann die Möglichkeit das erzeugte PWM-Signal von einem der Ausgänge auf den Motor zu schalten oder am Meßgerät zu beobachten.

Um Änderungen am Quelltext vorzunehmen müssen Sie den Debugger wieder verlassen.

7. Kurze Zusammenfassung

Der Takt f_{cc6} wird durch den in **CCU6_TCTR0** angegebenen Vorteiler geteilt. Die Betriebsart (Edge-/Center Aligned Mode) wird ebenfalls in **CCU6_TCTR0** eingestellt.

Die PWM-Frequenz ergibt sich aus dem eingestellten Takt und dem Wert in **CCU6_12PR**.

Die Betriebsart der Ausgänge wird in **CCU6_T12MSEL** eingestellt.

Welche Ausgänge invertiert sind wird in **CCU6_PSLR** eingestellt.

Das Tastverhältnis der Ausgänge wird über die Register **CCU6_CC6xR** eingestellt. Zum Beschreiben dieser Register werden die Shadow-Register **CCU6_CC6xSR** benötigt.

ALLE für diesen Versuch benötigten Informationen sind in dieser Anleitung enthalten.

8. Beispielprogramm

```
Void main (void)
{
  ALTSEL0P1L = 0x0003;
  DP1L = 0x0003;           // Ausgänge P1L.0 und P1L.1 für CC6
  CCU6_T12PR = 1000;      // Timer zählt bis 1000
  CCU6_TCTR0 = 2;         // vorteiler durch 4
  CCU6_CC60SR = 500;      // Tastverhältnis 50%
  CCU6_T12MSEL = 0x0003; // Compare-Mode
  CCU6_PSLR = 0x0001;     // CC60 (P1L.0) ist invertiert
  CCU6_MODCTR = 0x0003;  // CC60 und COUT60 auf Ausgang
  CCU6_TCTR4 = 0x0040;   // Shadow-Transfer
  CCU6_TCTR4 = 0x0002;   // Start
  while(1);              // definiertes Ende
}
```

Dieses Beispielprogramm erzeugt ein PWM-Signal mit einer Frequenz von 10 KHz an den Ausgängen P1L.0 und P1L.1. Einer der beiden Ausgänge ist invertiert.

9. Aufgabenstellungen

Sie erhalten die genauen Parameter der Aufgabenstellung zu Beginn des Praktikums. Jeder Teilnehmer erhält eine unterschiedliche Aufgabenstellung. Um sich auf das Praktikum vorzubereiten können Sie folgende Beispielaufgaben bearbeiten:

- Bestimmen Sie die für folgendes PWM-Signal erforderlichen Registerwerte.
PWM-Frequenz: 5 KHz
Edge Aligned Mode
Signal mit Tastverhältnis 2:1 auf CC60 ausgeben
Invertiertes Signal auf COUT60 ausgeben
Invertiertes Signal mit Tastverhältnis 4:1 auf COUT62 ausgeben.
- Welche Frequenz und welches Tastverhältnis hat das ausgegebene Signal?
CCU6_TCTR0 = 6
CCU6_T12PR = 375
CCU6_CC60R = 125
- Wie verändert der Wechsel von Edge Aligned auf Center Aligned Betriebsart die erzeugte PWM-Frequenz?

Gehen Sie bei Ihren Berechnungen von einer Frequenz f_{cc6} von 40MHz aus. Wenn der Vorteiler in **CCU6_TCTR0** nicht vorgegeben ist können Sie hier einen beliebigen Wert annehmen.