**2023\_SoSe\_Inf2\_2.7\_MCT\_TimerCapture,Aufgabe6(12.04.2023)**

0:02  
Wir haben uns letzte Woche den Timer angeschaut.

0:07  
Und das ist ja die Grundfunktion vom Timer, die wir gerade auch programmiert haben.

0:16  
Hier haben wir unser Zellregister. Das ist der Kern des Ganzen. Also ein Timer ist einfach ein Zähler, der.

0:24  
Zeit zählt.

0:28  
Die Zeit kriegen wir von einem internen Takt, der wird irgendwo am Prozessor bereitgestellt und wird dann hier rein gefüttert.

0:36  
Wir können den Timer ein und ausschalten, das ist hier symbolisiert durch dieses Run Flag. Wenn wir hier unterbrechen, kriegt der Timer keinen Takt und dann bleibt er stehen.

0:46  
Und wir können den Timer verteilen, indem wir da einen Vorteil auswählen.

0:52  
Das haben wir hier gemacht. 19 oder 83. Dann haben wir.

0:58  
Einen größeren Bereich, den wir dann abdecken können.

1:04  
Wir können den Zähler auch an externe Ereignisse anschließen. Das werden wir auch machen in einem.

1:11  
Versucht, da werden wir eine Frequenz messen, da kriegen wir ein Rechtecksignal von außen und messen dann die.

1:19  
Frequenz.

1:23  
Wir könnten auch ein Drehkreuz anschließen und messen, wieviel Leute im Stadion zum Beispiel jetzt das Stadion betreten haben.

1:30  
Also.

1:33  
Können wir zählen?

1:35  
Einfach ein ganz normaler Zähler. Wir können die Takts zählen.

1:43  
Dann haben wir eine Zeit oder wir können Ereignisse zählen, dann wissen wir, wieviel Ereignisse passiert sind, wieviel Leute ein Stadium oder ein Konzertsaal betreten haben oder?

1:54  
Was auch immer uns interessiert an der Außenwelt.

1:57  
Das ist der Zähler, der zählt zählt. Manche Zähler zählen hoch, andere zählen runter.

2:03  
Unser Zähler zählt jetzt hoch.

2:07  
Ist manchmal konfigurierbar.

2:10  
Das ist einmalig. Wir wollen jetzt aber im periodischen Zähler haben und deshalb.

2:16  
Haben wir noch ein Zusatzregister und eine Variante ist unser Zähler zählt hoch. Wir vergleichen kontinuierlich mit dem Zusatzregister, das macht die Hardware für uns, da müssen wir nichts programmieren, wenn das Mal konfiguriert ist, dann passiert das von alleine.

2:31  
Und wenn.

2:33  
Der Zähler des den Wert vom Zusatzregister erreicht hat. Dann wird der Zähler wieder auf 0 zurückgesetzt und wir kriegen ein Interrupt.

2:41  
Und dann passiert das Ganze von vorne. Der Zähler zählt wieder hoch und.

2:46  
Wenn er wieder oben ist, an dem Wert angelangt ist, dann kriegen wir wieder ein Interrapt.

2:52  
Also können wir einen periodischen.

2:54  
Timer installieren und mit Vorteile und Zusatzregister haben wir da sehr große Flexibilität.

3:02

3:05  
Das haben wir das letzte Mal besprochen und das ist ja auch Teil der ersten Aufgabe.

3:10  
Mit dem Zähler können wir aber noch mehr machen. Eine andere Funktionalität ist die.

3:15  
Funktionalität.

3:18  
Und zwar geht es darum.

3:21  
Signale zu auszuwerten.

3:27  
Das werden wir dann in der Frequenzmessung machen.

3:48  
Das ist ein Rechtecksignal.

3:51  
Dass wir jetzt vermessen wollen. Und wenn wir von dem Signal die Frequenz messen wollen, dann müssen wir diese Zeit hier bestimmen.

4:01  
Und diese Zeit ist dann proportional, also 1. Durch diese Zeit ist proportional zur Frequenz.

4:16  
Wenn wir die Zeit kennen, wissen wir f.

4:21  
Ist gleich 1 durch Delta.

4:24  
T.

4:27  
Wie können wir diese Zeit messen?

4:33  
Welches Register?

4:42  
Nehmen wir mal an, das wäre hier.

4:46  
ZB.

4:49  
GPIOD.

4:53  
Pin ich nehme jetzt mal irgendeinen, die haben wir alle im Display, sind d noch einer frei?

5:00  
Nehmen wir GPIUAPIN.

5:03  
1.

5:05  
GPUAP 0 ist ja der Taster.

5:09  
Nehmen wir mal, wir hätten jetzt diesen Pin hier.

5:13  
Dann würde das Signal sein.

5:16  
Wie würde jetzt ein Code ausschauen, mit dem wir dieses Signal vermessen würden?

5:23  
Sie haben ja gesagt, wir fragen den PIN ab, wie könnte das ausschauen?

5:31  
Richtig.

5:36  
Wir müssen jetzt aber hier den Wert abfragen.

5:41  
Und hier?

5:48  
Das Programm so einstellen, dass es auch ne steigende reagiert und dann im Timer startet, der sich wieder beendet, sobald die nächste die nächste steigen kann.

5:57  
Das ist schon die Kür.

6:01  
Wenn ich jetzt das Programmieren müsste, dann müsste ich erwarten, bis ich hier bin. Das könnte ich machen, indem ich sage, weil.

6:11  
Gpioa.

6:14  
IDR und.

6:18  
1 shift left 1.

6:24  
Ist gleich 0.

6:32  
Da bin ich jetzt hier.

6:34  
Ich frage den PIN ab.

6:36  
Ich muss den maskieren. Also ich möchte nur den PIN 1 wissen, nicht das ganze IDR Register sind ja 16 pins, ich möchte nur den pin 1 wissen, deshalb maskiere ich hier mit dem 1 und schaue ist das 0.

6:48  
Solange das 0 ist, bin ich hier. Wenn dieser PIN auf 1 geht.

6:55  
Dann ist das nicht mehr so. Dann bricht diese Schleife ab.

7:02  
Und dann könnte ich jetzt hergehen und sagen, Timer 1 ist gleich.

7:08  
Teil 7

7:10  
Count. Da lese ich das.

7:12  
Countregister aus.

7:20  
Dann habe ich jetzt genau hier.

7:23  
Das Countregister ausgelesen.

7:31  
Und jetzt muss ich warten, bis das nächste Mal hier kommt.

7:37  
Das heißt, jetzt ist unser PIN 1.

7:41  
Ich warte also, solange der hier 1 ist, weil.

7:46  
GPIOAIDR und.

7:51  
1 shift left 1.

7:56  
Ungleich 0.

8:05  
Da ist der jetzt 1. Dann geht er wieder auf 0. Dann warte ich wieder.

8:27  
Und jetzt ist die Tafel zu Ende.

8:35  
Wenn hier warte ich so, solange der PIN 1 ist. Das ist oben im Rechteck, hier warte ich solange der PIN 0 ist. Das ist unten im Rechteck und jetzt hab ich die nächste Flanke. Jetzt bin ich hier.

8:50  
Und da kann ich dann sagen Delta t.

8:54  
Ist gleich Timer 7.

8:58  
Countregister minus.

9:03  
1 und die 2 Variablen muss ich natürlich vorher anlegen.

9:08  
U. In 16 TT 1.

9:13  
Delta T.

9:18  
So in etwa könnte ein Code ausschauen, da muss ich noch n bisschen Feintuning machen und ich würd hier vielleicht auch ne Schleife machen, weil das wiederholt sich ja das hier und das hier und da muss ich am Anfang noch aufpassen, aber das ist die Idee, ich warte.

9:33  
Solange ich 0 bin, wenn der, wenn ich nicht mehr 0 bin, dann ist der PIN auf 1 gegangen. Dann habe ich meine Flanke, dann nehme ich die Zeit, merk mir die.

9:42  
Dann warte ich, solange ich auf 1 bin. Ich warte solange ich auf 0 bin und wenn der dann wieder auf 1 geht, dann lese ich wieder den Timer aus.

9:53  
Und bildet die Differenz. Ich ziehe den alten Wert ab und dann habe ich mein Delta t. Das ist genau die Zeit dazwischen und daraus kann ich dann f berechnen.

10:07  
So könnte ein Code ausschauen. Was ist der Nachteil von dem Code?

10:15  
Ja, der blockiert die ganze Zeit steht mein Programm an diesen Stellen, weil das sind ja busy waiting schleifen.

10:24  
Ich warte ja aktiv da in diesen Waldschleifen, bis von außen was passiert.

10:30  
Und das ist blöd.

10:34  
Und damit wir nicht busyvaking machen müssen, bietet der Timer uns die Möglichkeit, das eleganter zu machen.

10:47  
Und zwar indem wir unseren Timer erweitern und hier unten noch sogenannte Capture Register.

10:54  
Hinzufügen.

10:59  
Oben ist der Timer, wie wir ihn gerade schon gehabt haben. Wir haben den Vorteil.

11:04  
Dann das Zellregister und das Zusatzregister, was dann die periodische Macht.

11:11  
Capture Register ist jetzt.

11:13  
Ein Zusatzregister und hier haben wir eine Kopiereinheit.

11:21  
Dieses Kästchen hier bedeutet, wir können den Wert von hier darunter kopieren.

11:28  
Wenn nämlich hier an dem PIN was passiert und das was an dem PIN passiert, das können wir hier in diesem Modul konfigurieren, in dem Flankendetektion.

11:38  
Wir können zum Beispiel sagen, wenn hier eine positive Flanke detektiert wird, also vorher 0 und anschließend 1.

11:48  
Dann löse diesen Kopiervorgang aus. Dann wird das Zellregister in das Capture Register kopiert.

11:56  
Das bietet uns das Timer Periphere im Capture Mode.

12:03  
Also ein Zusatzregister wo wir was speichern können und eine Kopiereinheit, die von dem PIN getriggert werden kann.

12:13  
Der Vorteil ist, das macht es das Timer Periphere alleine, ohne dass wir was machen müssen.

12:21  
Wir müssen es nur richtig konfigurieren.

12:24  
Dann hätten wir.

12:27  
Diesen Teil hier, das Warten, die Flankenerkennung, macht er alleine und das Kopieren von hier nach hier macht er auch alleine.

12:38  
Was er nicht kann, ist diese Differenz bilden. Wie können wir jetzt die Differenz lösen?

12:51  
Müssen wir ja.

12:56  
Bitte.

13:01  
Nee, das Register hier, wir fangen ja mit t 1 an.

13:07  
Und t 1 ist ja hier beim Start, wenn die erste Flanke auftritt.

13:12  
Also wir müssten dann bei der ersten Flanke das Register auf 0 setzen. Das ist nicht so einfach.

13:22  
Die Differenz berechnen wir einfach selber.

13:24  
Das kann das Periphere nicht, das müssen wir selber machen. Und wo berechnen wir die Frequenz?

13:38  
Beim Interract.

13:41  
Wir lassen uns ein Interrad geben, wenn eine Flanke aufgetaucht ist, dann kopiert er das hier runter und dann haben startet unser Interrupt. Der Interrupt startet nicht sofort, das dauert immer ein kurzes bisschen.

13:54  
Aber das reicht uns. Wir starten den Interrupt und im Internat machen wir 2 verschiedene Dinge im ersten Fall.

14:03  
Lesen wir uns den Wert aus, aber nicht aus dem Countregister, sondern wir haben ja das captcha Register.

14:10  
Denn wir kriegen ja den Interrap hier. Das Countregister ist ja weiter gezählt, aber hier steht der Wert drin, der Zeitwert, als die Flanke passiert ist.

14:21  
Den hat uns das Periffre zwischengespeichert, den lesen wir aus und speichern ihn in eine Variable.

14:29  
Also wir lesen da nicht Timer 7 count, sondern Timer 7 capture Register und etwas später das Busyvating. Brauchen wir nicht etwas später kommt die nächste Flanke, wir kriegen wieder ein Interrupt, dann lesen wir wieder das Capture Register aus und ziehen den alten Wert ab.

14:47  
Dann haben wir die Differenz und die Differenz ist genau proportional zur Frequenz.

15:01  
Hier ist das noch mal dargestellt. Die blaue Linie ist unser Zähler und der zählt hoch, hoch, hoch, hoch, hoch bis er überläuft, dann springt er wieder zurück auf 0.

15:15  
Hier passiert der erste Capture Event, dann wird der Zählerstand TC 1 gespeichert.

15:22  
Etwas später bei t 2 passiert der nächste captcha wert. Da wird der zählerstand t 2 TC 2 gespeichert.

15:30  
Und dann können wir die Differenz bilden. T 2 minus, t 1 und.

15:36  
Das Periphren hat ja Takte.

15:39  
Die Takte müssen wir dann noch in die Frequenz umrechnen und dazu müssen wir die Frequenz des Zählers berücksichtigen.

15:45  
Wenn unsere Zähler mit ein Megahertz läuft und wir haben 1000 Takte, dann sind das eine Millisekunde.

15:55  
Also die Zellfrequenz müssen wir noch reinrechnen, wenn wir die Zeit haben wollen.

16:00

16:42  
Also unser Zähler zählt hoch.

16:52  
Besser hier zurückspringt, das ist unser.

16:55  
Maximalwert.

16:58  
Der ARR.

17:01  
Hier passiert unser erstes Event.

17:06  
Dann kriegen wir hier den Zählerstand 1, weil dieser Wert hier rüberkopiert wird. Also das ist die Zeit.

17:14  
Hier kriegen wir unser Rechteck.

17:19  
Hier kommt die zweite Flanke.

17:28  
Dann kriegen wir hier den zweiten Zeitwert.

17:33  
Dann haben wir hier die Differenz.

17:42  
Jetzt bilden ziehen die 2 hier voneinander ab und dann wissen wir, wieviel Takte hier zwischen sind.

17:50  
Das sind Takte.

17:54  
Weil wir hier ja hochzählen von 0 bis fünfundsechzigtausendfünfhundert.

18:04  
Dann haben wir die Differenz in Takten. Das ist schon ziemlich gut. Diese Takte müssen wir jetzt noch umrechnen in die Zeit und dazu brauchen wir die Frequenz, mit der unser Zähler läuft, inklusive Vorteil.

18:22  
Also unser Interrapt.

18:29  
Und Capture machen wir mit Timer 8 glaub ich timer 8 interrupt.

18:46  
Den gibt es nicht. Der heißt anders. Wenn wir dann die Aufgabe machen. Ich nenne ihn jetzt mal so.

18:57  
Wir lesen hier.

18:59  
Das aus, wir sagen U. In 16 t.

19:05  
1.

19:06  
TC 1 ist gleich.

19:10  
Timer 8.

19:12  
Capture compare Register.

19:16  
1, so heißt das.

19:23  
Und den Interrupt kriegen wir ja hier.

19:27  
Und hier, das heißt, wir müssen ein bisschen Logik in den Interrupt einbauen.

19:33  
Wir müssen jetzt die Differenz bilden.

19:38  
Oder Delta T ist gleich TC 2 minus t.

19:49  
Das ist der neue Wert TC 2 TC 1 ist der alte Wert vom letzten Mal. Die 2 ziehen wir voneinander ab, das ist unser Delta t.

20:00  
Uns den alten Wert merken, das ist der hier.

20:09  
Das wäre jetzt ein ganz einfacher Interract, der.

20:14  
Beim Interrap den Wert nimmt.

20:19  
Also beim ersten Interrupt kriegen wir noch nichts Gescheites raus, da lesen wir den Counter ab. Wir bilden eine Differenz, die ist Unsinn und dann merken wir uns den Wert in TC 1 beim.

20:32  
Ah, das ist ein kleiner Trick. Das muss TC 2 sein und das hier.

20:39  
Müssen wir außerhalb definieren?

20:42  
Lesen hier TC 2 einziehen den früheren Wert ab, sofern wir einen haben?

20:50  
Und anschließend ist der aktuelle Wert.

20:54  
Der alte Wert fürs nächste Mal, den müssen wir in einer globalen Variablen speichern, außerhalb vom Interract, weil der muss zwischen 2 Interrupts müssen wir uns den merken. Die Variable hier gibt es nur im Interrap, die wird danach wieder gelöscht.

21:18  
Okay.

21:20

21:25  
Frage.

21:42  
Dasselbe Bild wie hier.

21:45  
Unser Interrapt läuft.

21:48  
Hoch.

21:50  
Und hier wieder runter.

21:52  
Und Hierhoch.

21:54  
Wir machen diese Capture so wie hier. Wir haben ein Rechtecksignal.

22:03  
Das Rechtecksignal schaut so aus.

22:11  
Machen wir ein bisschen.

22:19  
So schaut das Rechteck Signal aus was kriegen wir jetzt?

22:24  
Wir haben es eingestellt, dass wir immer auf der.

22:27  
Positiven Flanke einen Trigger kriegen, das heißt, Wir kriegen einen Trigger hier.

22:33  
Und kriegen hier unseren Wert. TC 1.

22:37  
Wo kriegen wir unser TC 2?

22:43  
Richtig, hier.

22:46  
Pc 2.

22:51  
Was ist jetzt unser Delta T?

23:00  
Vorher war es ja.

23:03  
TC 2 minus TC 1 jetzt haben wir ein Problem, wenn wir das machen.

23:10  
TC 2 ist kleiner als TC 1.

23:19  
Ja.

23:29  
Richtig, da müssen wir jetzt fünfundsechzigtausendfünfhundertsechsunddreißig dazuzählen.

23:41  
Es müßten 35536 sein.

23:48  
Weil wir nämlich in der Zwischenzeit einen Überlauf haben.

23:53  
Unser Zähler läuft hier einfach hoch und wir wissen ja nicht, wann wir diese Rechtecke kriegen. Die können irgendwie liegen, also kann es uns passieren. Wir haben eine Flanke vor dem Überlauf und die nächste Flanke nach dem Überlauf.

24:08  
Das führt dazu, dass uns hier fünfundsechzigtausendfünfhundertsechsunddreißig fehlen, weil der Zähler in der Zwischenzeit zurückgesetzt wurde auf 0.

24:17  
Das müssen wir korrigieren, indem wir das dazuzählen. Und wenn es ganz blöd kommt, wenn wir ein ganz flexibles System haben wollen.

24:34  
Mit einer sehr langsamen Frequenz.

24:56  
Dann müssten wir.

24:59  
Die Überläufe mitzählen.

25:05  
Dann müssten wir nämlich 3 mal fünfundsechzigtausendfünfhundertsechsunddreißig dazuzählen.

25:15  
Weil der Zähler ja immer zurückgesetzt wird. Jedes Mal, wenn er zurückgesetzt wird, müssen wir uns das merken und Fünfundsechzigtausendfünfhundertsechsunddreißig dazuzählen, einmal zurückgesetzt, zweite Mal zurückgesetzt, das dritte Mal zurückgesetzt.

25:28  
Und dann wäre eben die das Delta TTC 2 minus TC 1 + 3 mal fünfundsechzigtausendfünfhundertsechsunddreißig.

25:40  
Den Fall.

25:48  
Den Fall machen wir nicht.

25:54  
Den Fall allerdings, den brauchen wir.

25:59  
OK, das ist hier auf den Folien dargestellt. Das ist der einfache Fall, wenn wir keinen Überlauf haben.

26:10  
Das ist der Fall mit Überlauf und da müssen wir eben korrigieren.

26:17  
Also wieviel Überläufe haben wir?

26:24  
Da gibt es aber einen Trick, wo man sich das sparen kann.

26:30  
Wenn diese Variablen alle 16 Bit sind, was passiert dann, wenn sie 65 536 dazuzählen?

26:43  
Nix.

26:45  
Richtig. Das heißt, wenn alle Variablen 16 Bit sind, dann können Sie diese Korrektur sparen.

26:57  
Nehmen wir ein Beispiel.

27:03  
Wir kriegen hier.

27:06  
Nehmen wir hier mal, wir kriegen hier 30000 und hier kriegen wir.

27:16  
10000.

27:18  
Was ist 10000 minus?

27:22  
1000.

27:24  
Das ist natürlich -20000 wenn sie alle wissen. Wenn das jetzt aber alles 16 bit Variablen sind, also wenn wir u int.

27:35  
16 TTC 2 ist gleich 10000.

27:42  
TC 1 ist gleich.

27:45  
30000 haben.

27:48  
Delta T.

27:52  
Was ist dann delta? T ist gleich TC 2 minus TC 1.

28:01

28:09  
Delta T ist dann.

28:12  
Fünfundvierzigtausendfünfhundert 36.

28:22  
Das ist eine variable Anzahlt in 16 bit. Die geht von 0 bis fünfundsechzigtausendfünfhundertfünfunddreißig, was anderes kann die Variable nicht haben und sie kann keine negativen Zahlen haben.

28:34  
Wir werden die negative Zahl -20000 im zweier Kompliment haben und die in einer Anzahl variable speichern. Dann wird automatisch fünfundsechzigtausend im Prinzip dazu gezählt.

28:51  
Das können Sie sich überlegen, wenn Sie die Aufgabe machen, mit der Frequenzmessung, dann können Sie das verwenden, wenn man die richtigen.

29:00  
Wenn man die richtigen.

29:04  
Datentypen verwendet dann.

29:10  
Bitte Code noch mal einfacher.

29:16  
Ok, das ist der Fall mit Überlauf. Entweder wir machen Fallunterscheidung und zählen Fünfundsechzigtausend dazu oder wir machen die richtigen Datentypen, dann können wir uns das Sparen.

29:30  
Wenn wir mehr als eine Periode haben, dann können wir mehr als einen Überlauf kriegen. Dann müssen wir die Überläufe zählen. Das wird dann aufwendiger, dann müssen wir.

29:42  
Bei jedem Überlauf einen eigenen Interab installieren und dann können wir in den Interrupts die Überläufe zählen. Das System zählt die Überläufe nicht von alleine.

29:57  
Das ist dann im Prinzip die Aufgabe.

30:03  
Eine weitere Aufgabe.

30:05

30:26  
Nämlich die Aufgabe, 6 Frequenzmessen.

30:45  
Da sollen sie die Software so erweitern, dass sie die Frequenz eines externen Signals vermessen.

30:56  
Und zwar verwenden wir dazu Capture, aber wir verwenden den Timer 12.

31:09  
Das müssen Sie dann konfigurieren in Capture Mode und dann?

31:19  
Können Sie damit die Frequenz bestimmen? Und dazu gibt es ein Aufgabenblatt? Das machen wir vielleicht zuerst.

31:31  
Hat jemand von ihnen zufällig so einen Micro USB Kabel dabei?

31:51  
Ich probier's mal.

32:14  
Danke.

32:16

32:41  
So, das ist aufgabenblatt 8 terma Capture frequenzmessung. Jetzt schau ich mal, ob das auch das Richtige ist.

32:48

33:11  
Genau das ist das Richtige.

33:14  
Das ist unser Periphre für die Capture Messung. Also wir haben hier oben dasselbe wie vorher den Counter, den Pre scaler.

33:24  
Und den Takt mit dem internen Clock, das ist jetzt Timer 8 timer 12.

33:31  
Das ist so wie Timer 7, nur kann ein bisschen mehr. Er hat nämlich diese Capture compare Einheiten die compair Funktion machen wir nächste Woche wozu die gut ist. Jetzt brauchen wir dann die Capture Einheit und die Capture Einheit kann jetzt mit einem PIN verbunden werden.

33:49  
Wir können jetzt also einen PIN hernehmen und sagen, schau dir das Signal an, dann haben wir einen Edge Detektor. Das ist eine Flankenerkennung.

33:58  
Und wenn eine Flanke erkannt wird, dann löst der hier ein Capture Event aus. Das heißt das wird hier runter kopiert und anschließend kriegen wir ein Interrupt.

34:09  
Hier oben raus.

34:12  
Also für die Capture für die Frequenzmessung verwenden wir die Capture Einheiten des.

34:22  
Des Timers und müssen das konfigurieren. Und das ist in dieser Aufgabe 8 zu konfigurieren. Wir haben wieder die üblichen Kandidaten, die wir gerade schon programmiert haben, wie Prescala, Countregister und autoreload Register.

34:37  
Das ist dasselbe wie in Timer 7. Und dann haben wir.

34:44  
Ein Kontrollregister, dass wir einschalten müssen. Und dann haben wir neue Register. Das ist das Capture compair enable Register.

34:56  
Da können wir bei dem CCMR 1.

35:06  
Können wir Verschiedenes einstellen?

35:09  
Wir können wählen, ob es n.

35:13  
Output ist, das machen wir jetzt nicht.

35:16  
Ob es n Input ist Zuordnung.

35:19  
YY oder input Zuordnung vertauscht und wir können keinen Vorteiler verwenden. Wir könnten nur hier noch einen Vorteiler machen, den brauchen wir aber nicht.

35:32  
Und wir können einen Filter anwenden. Das machen wir aber auch nicht. Also Filter machen wir 0 prescala machen wir 0 und wir können die Selection machen. Die Selection ist folgendes.

35:43  
Wir können, wenn wir hier reingehen, mit dem PIN.

35:48  
Können wir entweder den hier triggern, den Capture oder den hier triggern.

35:54  
Das können wir über diese Selektoren hier auswählen. Wir sagen einfach, wir nehmen den PIN, triggern den Inputfilter, gehen in den Pre Scanner und machen dann hier ein Captcha.

36:19  
Das ist die erste Konfiguration von dem CCMR 1.

36:24  
Und da kann ich 2 Kanäle machen.

36:28  
Das hier ist der Kanal 1 CC 1.

36:33  
IC 1 p prescala.

36:36  
IC 1 Filter also wenn ich den Kanal 1 nehme, dann muss ich hier 0000 reinschreiben, das ist der Filter den Prescala verwenden wir nicht 00 und dann die Zuordnung 01.

36:50  
Wenn ich den anderen PIN nehmen will, dann muss ich in die obere Hälfte von dem Register das reinschreiben, dann hätte ich hier meinen Filter, hier meinen Prescala und hier meinen Selektor.

37:02  
Das brauchen wir aber nicht. Wir haben in dem Fall nur einen PIN und den verwenden wir direkt mit dem Capture, also das Register, das Modem Register hat 2 Teile den hier und den hier, davon brauchen wir nur einen Teil.

37:17  
Dann müssen wir das Enable Register aktivieren. Das ist dieses hier, das hat auch 2 Teile, nämlich diese 4 bit und diese 4 bit, wenn wir den ersten PIN haben, dann interessieren uns nur diese 4 bit hier die unteren 4 und da haben wir 4 einzelne 3 einzelne Bits die wir setzen, nämlich das hier, das ist das Enable, das müssen wir auf 1 setzen, damit überhaupt etwas passiert und dann haben wir die 2 hier.

37:47  
Cc 1 NP und CC 1 p. Die sind jetzt dummerweise hier bei 1 und 3.

37:54  
Wenn die 200 sind, dann haben wir eine steigende Flanke, dann reagiert er auf eine steigende Flanke. Wenn die 201 sind auf eine fallende und bei 10 würde auf steigende und fallende Reagieren, dann würden wir also bei steigender und fallender einen Trigger kriegen.

38:11  
Wollen wir aber in dem Fall nicht.

38:15  
Das müssen wir konfigurieren.

38:21  
Die captcha Werte sind dann in den Registern.

38:24  
CCR 1 und CCR 2 zu finden.

38:32  
Und dann haben wir unser Diener wieder, unser DMA.

38:37  
Interrupt enable Register. Da haben wir jetzt aber mehr Flex als das letzte Mal. Das letzte Mal hatten wir nur den Überlauf, wenn der Timer eine Periode abgelaufen ist, haben wir n Interrap gekriegt, alle Millisekunde, den brauchen wir jetzt hier nicht, weil die Periode interessiert uns nicht, wir brauchen die Capture Events, das wäre der PIN 1 CC 1 enable das wäre der PIN 2. Wir arbeiten nun mit PIN 1 und nehmen dafür.

39:06  
Den hier.

39:09  
Da können wir ne 1 reinschreiben, dann kriegen wir ein Interrupt und dann müssen wir natürlich im Nvig wieder. Genau dann haben wir ein Statusregister.

39:19  
Da haben wir jetzt auch ein bisschen mehr. Das ist der Überlauf, den müssen wir wieder auf 0 setzen.

39:25  
AD beziehungsweise wir können ihn ignorieren, den Interrap, das wäre das Capture Flag, das müssen wir in der Software auch auf 0 setzen.

39:33  
Aber es wird automatisch auf 0 gesetzt, wenn wir.

39:38  
Das Capture Register auslesen.

39:43  
Dann müssen wir das Periphere einschalten. In dem Fall ist es periphere 12 hier.

39:49  
Das ist Timer 7, das ist Timer 12 direkt daneben, kein großer Unterschied.

39:57  
Und dann brauchen wir noch.

40:01  
Die Envig.

40:04  
Und dieser Interrupt heißt jetzt ein bisschen anders. Der hat nämlich einen komischen Namen, das ist Timer. 8 unterstrich Brigg unterstrich Timer, 12 unterstrich ihre QN.

40:16  
Weil es nur ein Internet gibt für Time 8 und Timer 12 zusammen, so heißt die Nummer von dem Interrupt und hier hinten steht dann der Händler, nämlich der hier Timer 8 birgt Timer 12 Interrupt ihre Kuhhändler.

40:33  
Das ist der Interrap, den wir dann installieren müssen.

40:46  
Genau das brauchen wir zur Frequenzmessung. Ah, eine Sache ist noch unklar für den Versuch nächste Woche.

41:02  
Welcher PIN ist das denn?

41:19  
Also wir wollen Timer 12 verwenden, das heißt, das ist der PIN team 12.

41:24  
Channel 1.

41:26  
Wenn sie den Prozessor anschauen, also das Ding an sich beziehungsweise die Beschriftungen von den Pins, dann wenn sie diesen pin nicht finden.

41:37  
Eigentlich hat der nur GPI OS zur ein und Ausgabe.

41:41  
Jetzt wollen wir hier aber n Timer haben und der Timer soll auf diesen PIN reagieren. Also wenn wir da am PIN eine Flanke kriegen, dann möchten wir da das was passiert.

41:55  
Wie können wir rausfinden, welcher PIN das ist?

42:05  
Ich sage es Ihnen natürlich.

42:08  
Wir haben ja die ganzen GPI OS.

42:11  
Und die haben ja schon programmiert. Leuchtdiode an, Leuchtdiode aus, Taste eingelesen, Taste ausgelesen, dieser Pin hier ist auch nichts anderes als ein spezieller GPIO.

42:22  
Damit wir rausfinden, welcher das ist, brauchen wir noch eine andere Dokumentation.

42:28  
Ich geh jetzt mal in die Originaldokumentation.

42:34  
In der Dokumentation und Moodle ist es gibt es die einfache Variante.

42:39  
Und das müsste die hier sein.

42:46  
Das ist jetzt nicht das Handbuch mit den tausendsiebenhundert Seiten. Das sind nur 185 Seiten, das ist nämlich diese eigentliche Hardwarebeschreibung des Pins und da, wenn ich jetzt einfach mal durch Scrolle, ohne die Details anzuschauen.

43:08  
Dann hoffe ich, dass ich im richtigen Dokument bin.

43:12  
Da ist beschrieben, welche Spannungen der hat, wie die Pegel an den.

43:16  
Pins sind und so elektrisches Zeug.

43:24  
Hier ist das PIN out.

43:26  
Das sind die Pins, die unser Prozessor hat.

43:30  
Aber es ist das falsche Package. Wir haben ein anderes Package, also würde das dann ausschauen, das müsste der sein, den wir haben, ja.

43:41  
Das ist der Prozessor, da sind die Pins.

43:44  
Gibt es verschiedene?

43:47  
Und.

43:58  
Hier fängt es jetzt an.

44:01  
Das ist jetzt die spezialfunktionen Pin Tabelle für jeden PIN den unser Prozessor hat ist hier aufgeschrieben welche Spezialfunktionen der hat, fängt an mit PIN.

44:13  
PA 0 und da steht zum Beispiel Timer 5, Channel 1. Wenn wir Timer 5 nutzen würden, dann wär das der PIN PA 0.

44:25  
Pin, PA, 14 und 15.

44:33  
Nein, die sind hier nicht gelistet. OK, jetzt muss man in dieser Tabelle also sozusagen suchen, Tim.

44:40  
11 channel 1.

44:43  
Und.

44:47  
Das hat der Kollege dankenswerterweise markiert.

44:53  
Hat er nicht.

44:58  
Ich weiß selber gar nicht, welcher PIN das ist. Das ist irgendwie.

45:13  
Hier die sind das ja da.

45:17  
Pin, PB, 14 und PB 15.

45:20  
Haben verschiedene Spezialfunktionen.

45:24  
Und hier haben wir Timer 12 channel 1 timer 12 channel 2 also wenn wir Timer 12 channel 1 nutzen wollen, dann müssen wir dazu.

45:33  
Den PIN.

45:38  
PB 14 nehmen also das hier ist.

45:45  
Der PIN.

45:47  
Und hier haben wir die Funktion Timer 1 ja.

45:55  
Nein.

45:57  
Direkt. Man kann den PIN dann direkt verschalten mit dem Timer dazu muss man allerdings die GPI OS konfigurieren, nämlich in Alternate function Mode.

46:09  
Alternate function mapping wenn ich nämlich hier nachschaue, dann hat dieser pin Timer 12 channel 1. Diese Funktion ist die Funktion AF 9 Nummer 9 die alternate function nummer 9. Ich muss also den PIN PB 14 in den Alternate Function Mode versetzen und dazu den Mode 9 eintragen.

46:35  
Wenn wir nochmal zurückgehen.

46:37  
In das Handbuch.

46:53  
Das war das Handbuch, da war die GPIOS alle und da gibt es hier den Alternate function Mode.

47:01  
Und zwar gibt es da 2 Register, weil Alternate Functions gibt es pro pin 16.

47:09  
Dafür brauchen wir 4 Bit, um die Nummer einzutragen.

47:14  
4 Bit pro pin Wir haben 16 pins, es sind 64 Bits und deshalb wurde das hier in 2 Register aufgeteilt, die heißen hier.

47:24  
GPIO low Register, Alternate function, Register Low und Alternate Function Register High. Im Programmcode ist das n Array mit 0 und 1. Das sehen Sie dann, wenn Sie es.

47:37  
Programmieren die Low Register sind für die unteren 8 Pins zuständig und wenn sie da hinschauen, dann steht hier.

47:45  
Alternate function Register Low 01234567 das sind die unteren 8 Pins 0 bis 7 und das High Register ist zuständig für die oberen 8.

47:59  
819 11.

48:03  
1314 15 unser pin war PB 14.

48:09  
Port GPIOB 14 das wäre dieses Special function Register.

48:15  
Und da müssen wir eine 9 reinschreiben, damit wir die Special function AF 9 kriegen.

48:27  
Also hier über diese AFR Register können wir im GPIO sagen, wähle diese Funktion und wir müssen natürlich, das hab ich jetzt vergessen im Mode Register das auswählen, aber wenn wir ins Modregister zurückgehen, das kennen wir ja schon 00 das input 01 ist general purpose output mode 10 ist alternate function mode für diesen pin.

48:50  
Also damit wir diese Capture aktivieren können, müssen wir GPIOB für den PIN 14.

48:58  
In alternate function Mode schalten 10, dann wird dieser pin nicht, ist mir nicht mehr als normales GPIO verwendbar, sondern er ist an eine andere Funktion gekoppelt.

49:10  
Und dann müssen wir noch sagen, welche Funktion, indem wir hier unten an die entsprechende Stelle.

49:18  
Einen neuen reinschreiben?

49:19  
Nehme ich an, die Funktion Nummer 9 und das ist dann timer 12 channel 1.

49:31  
Also das ist etwas komplizierter. Als Vorbereitung gebe ich Ihnen dieses Übungsblatt mit für nächsten Dienstag, das hier freigeschaltet ist. Aufgabenblatt 8 Timer Capture bereiten Sie das vor.

49:45  
Das gehen wir dann in der Vorlesung durch und so wie sie es jetzt ergeben hat, müssen wir dann den Versuch 6 nächste Woche machen. Der Versuch 5 kommt dann die Woche drauf, also wir vertauschen versuche 5 und 6.

49:59  
Weil 5 ist comper Funktion, die haben wir jetzt noch gar nicht angeschaut.

50:05  
OK, das war es für die für heute und für diese Woche nachher Versuch und morgen Versuch mit den Timer 7 ist easy, jetzt haben sie ja schon gesehen was als Nächstes kommt, Timer 7 ist dagegen easy mit den Leuchtdioden blinken und so, das muss man ein bisschen frickeln nächste Woche dann capture, da werden wir eine Frequenzmessung machen und.

50:29  
Den Capture, das ist dann sprechen wir dann nächsten Dienstag nochmal und machen wir dann nächste Woche. Ansonsten wünsche ich Ihnen noch eine schöne Woche und wir sehen uns im Praktikum dann nachher.