

Bericht Semesterarbeit

Fingerprint Authentication System

Autoren:

Betreuender Dozent:

Fach:

Abgabedatum:

Marc Plaz, Andreas Lang

Prof. Dr. Guido M. Schuster

Digitale Signalverarbeitung

13. Februar 2004

Abstrakt

Diese Arbeit beschäftigt sich mit dem Einlesen und dem Vergleichen von Fingerabdrücken. Das Einlesen der Fingerabdrücke wurde mit dem Fingerabdrucksensor FPC1010, der als Daughter-Card zum Texas Instruments DSP TMS320C6711 DSK erhältlich ist, realisiert. Durch die McBSP-Schnittstelle erreicht man schnelle Transfers von Daten zwischen der Daughter-Card und dem DSK-Board. Das Vergleichen der Fingerabdrücke war die Hauptaufgabe.

In dieser Semesterarbeit haben wir eine Art Korrelationsverfahren benutzt um zwei Fingerabdrücke(kurz FP) zu vergleichen. Wie sich herausstellte war unsere Methode mit sehr vielen for-Schleifen, riesigen Datenmengen und begrenztem internen Speicher verbunden.

Der Knackpunkt bei dieser Aufgabe war es, das Vergleichen möglichst in Real Time zu gestalten. Um den Vergleichs-Algorithmus schnell zu halten war es von Nöten die Rechnungen möglichst nur im internen Speicher (schnelle Zugriffszeiten) zu machen. Da nicht alles was man zum Rechnen brauchte, gleichzeitig im IRAM Platz hatte, mussten wir uns etwas einfallen lassen. Es boten sich zwei Möglichkeiten an: Entweder man komprimiert die Bilder zu Templates mit weniger Speichergrösse oder man benutzt QDMA (Quick Direct Memory Access) Blocktransfers.

Wir entschieden uns in einem Gespräch mit unserem Betreuer, Herrn Schuster, für die QDMA Blocktransfers.

Letztlich wurde ein Algorithmus entwickelt, der in Real Time einen Fingerabdruck mit drei anderen vergleicht.

Inhaltsverzeichnis

1. EINLEITUNG	5
1.1 AUFBAU	5
1.2 ERGEBNISSE	5
1.3 AUSBLICK	6
2. AUFGABENSTELLUNG	7
2.1 KURZBESCHREIBUNG	7
2.2 AUFGABENSTELLUNG	7
2.3 ERWARTETE ERGEBNISSE.....	7
2.4 ARBEITSWEISE.....	7
3.GRUNDLAGEN FINGERABDRUCK- ERKENNUNGS-SYSTEM	8
3.1 EINFÜHRUNG	8
3.2 DAS GRUNDPRINZIP	9
3.3 DIE 4 BASEELEMENTE EINES FINGERABDRUCK-ERKENNUNGS-SYSTEM	9
3.3.1 <i>Sensor Element</i>	10
3.3.2 <i>Prozessing Element</i>	11
3.3.3 <i>Speicher Element</i>	13
3.3.4 <i>Interface Element</i>	13
3.4 GENAUIGKEIT VON BIOMETRISCHEN METHODEN.....	14
4. LÖSUNGSVARIANTEN IN MATLAB.....	16
4.1 ÜBER MATLAB	16
4.2 DIGITALE BILDDARSTELLUNG	16
4.3 PRE-PROZESSING	17
4.3.1 <i>Filter</i>	18
4.3.2 <i>Filterversuch</i>	19
4.3.3 <i>Der Matlab Code des Filters</i> :.....	19
4.3.4 <i>Darstellung des vom Sensor gelieferten Fingerabdrucks auf Matlab</i>	20
4.4 MATCHING-ALGORITHMUS	21
4.4.1 <i>Funktion</i>	21
4.4.2 <i>Matching 1</i>	21
4.4.3 <i>Matching 2</i>	22
4.4.4 <i>Schlussfolgerung</i>	22
4.4.5 <i>Schwellwert</i>	23
4.4.6 <i>Matlab-Code des Matching-Algorithmus</i> :	24
5 ENTWICKLUNGSUMGEBUNG	25
5.1 TMS320CD6711 DSK	25
5.2 FPC1010 FINGER PRINT SENSOR DAUGHTER CARD.....	26

5.3 CODE COMPOSE STUDIO	28
6 INBETRIEBNAHME DER DAUGHTER CARD...	30
7 ECHTZEITIMPLEMENTIERUNG.....	31
7.1 SPEICHERBELEGUNG DES 6711 DSK.....	31
7.2 GEL (GENERAL EXTENSION LANGUAGE)	32
7.3 DATENÜBERTRAGUNG	33
7.3.1 <i>McBSP</i>	33
7.3.1 <i>EDMA</i>	35
7.3.3 <i>Emif</i>	37
8 SCHLUSSWORT.....	39
9 VERZEICHNISSE.....	40
9.1 ABBILDUNGEN.....	40
9.2 TABELLEN	40
9.3 FORMELN	40

1. Einleitung

In unserer ersten Semesterarbeit an der Hochschule Rapperswil Abteilung Elektrotechnik haben wir uns im Fach Digitale Signalverarbeitung für die Semesterarbeit Fingerprint Authentication mit dem DSK TI C6711 entschieden. In dieser Arbeit geht es grundsätzlich darum, Personen zuzulassen, deren Fingerabdruck dem System bekannt sind. Die genauere Aufgabenstellung wird unter Punkt 2 erläutert.

1.1 Aufbau

Unser Bericht ist in 4 Abschnitte gegliedert:

- Grundlagen des Fingerabdruck-Erkennungs-Systems.
- Erste Versuche des Pre-Prozessing und Matching-Algorithmus auf Matlab
- Inbetriebnahme der Daughter-Card und Hardwarebeschreibung der Hauptkomponenten
- Echtzeitimplementation Fingerprinterkennung

1.2 Ergebnisse

Unser Fingerprint Authentication System ist ein wenig anfällig auf verschiedenen Druck und Rotation. Die Empfindlichkeit auf Druck konnten wir durch einen Preprozess-Filter ein wenig mindern. Dennoch bei sehr unterschiedlichem Druck zeigt unserer Erkennungs-System Fehler auf. Auch bei der Rotation sind grössere Probleme aufgetaucht als erwartet. Wir haben dieses Problem zwar mit drei verschieden positionierten Templates kompensieren wollen, allerdings war der Effekt nicht genügend, da nur schon kleine Abweichungen in der Rotation zu ungenauen Ergebnisse des Matchings führt

Die Verarbeitungszeit konnten wir durch die Verwendung vom internen Speicher und dem Enhanced Direct Memory Access auf eine akzeptierende Zeit hinunter bringen.

1.3 Ausblick

Wenn wir diese Arbeit weiterführen würden, wäre es unserer Meinung nach sinnvoll, einen anderen Matching-Algorithmus zu benutzen. Die minuten-basierende Erkennung oder die Mustererkennung sind dafür bewährte Lösungsansätze. Wir würden uns sehrwahrscheinlich für die minuten-basierende Erkennung entscheiden, da es sich optimal für Systeme mit schnellen Rechnern eignet. Das DSK6711 stellt dafür, mit dem schnellen DSP-Prozessor die richtigen Voraussetzungen.

2. Aufgabenstellung

Thema: Fingerprint Authentication mit dem DSK TI C6711

Studenten: Marc Aron Plaz und Andreas Lang
Betreuer: Prof. Dr. Guido M. Schuster

2.1 Kurzbeschreibung

Die SA besteht darin, eine kommerzielle Daughter-Card für den DSK C6711 zu benutzen, um ein Fingerprint Authentication System zu bauen. Das Ziel der Arbeit ist es, nur Personen zuzulassen, deren Fingerabdruck dem System schon bekannt sind.

2.2 Aufgabenstellung

- Erarbeitung und Darstellung der erforderlichen Grundlagen
- Inbetriebnahme der Daughter-Card und der mitgelieferten Software
- Echtzeitimplementation Fingerabdruckerkennung

2.3 Erwartete Ergebnisse

- Dokumentation der Grundlagen
- Dokumentation des Problems und der Lösung(en)
- C Programm für den DSK
- Dokumentation der Implementierungen
- Abgabe eines Zwischenberichtes am 22. Dezember 2003 um 12:00

2.4 Arbeitsweise

- Sie führen ein persönliches Laborbuch, wo Sie aufschreiben wann Sie was für wie lange machen und was die Ergebnisse sind
- Sie schicken mir vor jedem Treffen eine Agenda und nach dem Treffen ein Protokoll

3. Grundlagen Fingerabdruck-Erkennungs-System

3.1 Einführung

Von allen biometrischen Systemen (Iris-, Gesichts-Erkennung, etc.) hat sich die Fingerabdruck-Erkennung als das Universellste durchgesetzt. Die verwendeten Fingerprint Input Devices sind heutzutage klein und kostengünstig. Der Markt hat schon frühzeitig das Potential dieses Gebietes erkannt und namhafte Firmen begannen mit der Herstellung von Fingerabdruck-Erkennungs-Systeme.

Fingerabdrücke sind ein unverwechselbares Kennzeichen eines jeden Menschen. Wie ja schon bekannt benutzt man die Fingerabdruckerkennung in diversen Gebieten. Wie zum Beispiel als Passwortsatz für das Login in Computersysteme, in der Gerichtsmedizin oder für die Zutrittskontrolle.

Vor allem bei der Zutrittskontrolle werden bei Neuinstallationen vermehrt Fingerabdruck-Terminals eingesetzt. Dabei ist die Hochsicherheit, die solche Systeme bieten eher zweitrangig. Dass solche Systeme ohne Schlüssel oder Karte auskommen wird als entscheidendes Argument für die Anschaffung angeführt.

Die Entwicklung von biometrischen Systemen ist ein Markt mit enormem Potential. Wie unten in der Abbildung¹ ersichtlich, wird von Jahr zu Jahr ein Umsatz in diesem Teilmarkt mit steigender Tendenz erwartet.

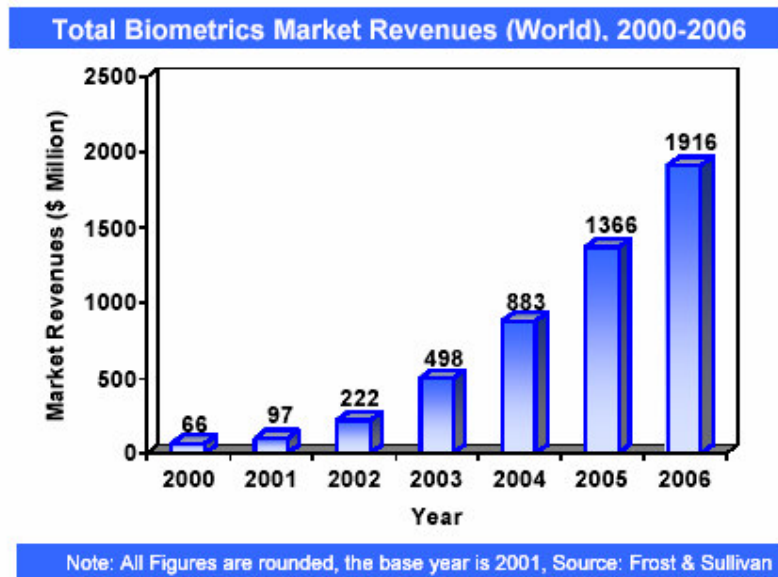


Abbildung 1 Biometrics Market Revenues¹

¹ Texas Instrument SPRA894A.pdf

3.2 Das Grundprinzip

Das Prinzip ist denkbar einfach zu verstehen. Eine Scannereinheit erstellt ein digitales Bild des Fingerabdrucks. Aus diesem Bild werden die wesentlichen Merkmale mittels eines Erkennungs-Algorithmus extrahiert und anschliessend gespeichert. Bei einer anschliessenden Prüfung, werden wiederum die Fingerabdruck-Bilddaten extrahiert und mit den gespeicherten Daten verglichen (Matching).

3.3 Die 4 Basiselemente eines Fingerabdruck-Erkennungs-System

Ein Fingerabdruck-Erkennungs-System besteht aus 4 Basisblöcken:

- Sensor
- Signal Processing
- Speicher
- Benutzeroberfläche

Anhand des Blockschemas (Abbildung 2) ist klar ersichtlich, wie die 4 Basisblöcke in einem System als Ganzes miteinander verbunden sind.

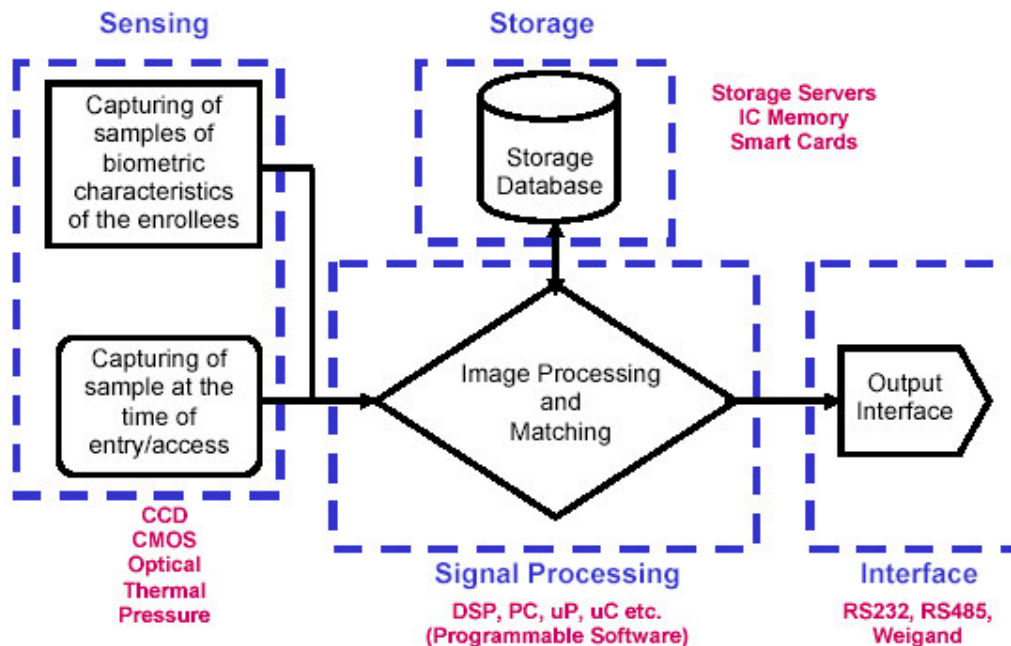


Abbildung 2 Basiselemente eines Fingerprintsystems ²

² Texas Instrument SPRA894A.pdf

3.3.1 Sensor Element

Dieser Block ist das Eingangs-Interface und auch die Kern-Hardware eines Fingerabdruck-Erkennungs-Systems. Der Sensor ist entweder eine opto-elektronische Einheit (CCD Kamera) oder ein kapazitiver Chip-Sensor, welcher heutzutage vermehrt zum Einsatz kommt, da er kostengünstiger und vor allem kleiner ist.

In unserem Fall wird genau so ein kapazitiver Chip-Sensor FPC1010 von Texas Instrument zur Anwendung kommen.

Der Sensor liest den anliegenden Fingerabdruck ein und liefert ein 2-dimensionales, 8-Bit Graustufen Bild, welches dann von dem Processing-Block weiterverarbeitet wird.

Anhand der Abbildung 3 ist zu erkennen wie so ein kapazitiver Sensor den Fingerabdruck im Detail interpretiert. Es handelt sich um einen sehr kleinen Ausschnitt des Sensors. Jede Zelle dieses Sensors liefert ein Pixel mit unterschiedlichem Grauton. So wird die Anzahl Pixel eines Digitalbildes anhand der Anzahl Zellen bestimmt. Die rote Kurve in der Abbildung 3 stellt eine ausgeschnittene Pixelreihe des Ausgangs dar. Hat die Kurve einen hohen Wert entspricht das einem Weiss, ist sie jedoch tief wird dieser Punkt als Schwarz gewertet, was einer Fingerabdruck-Rille entspricht.

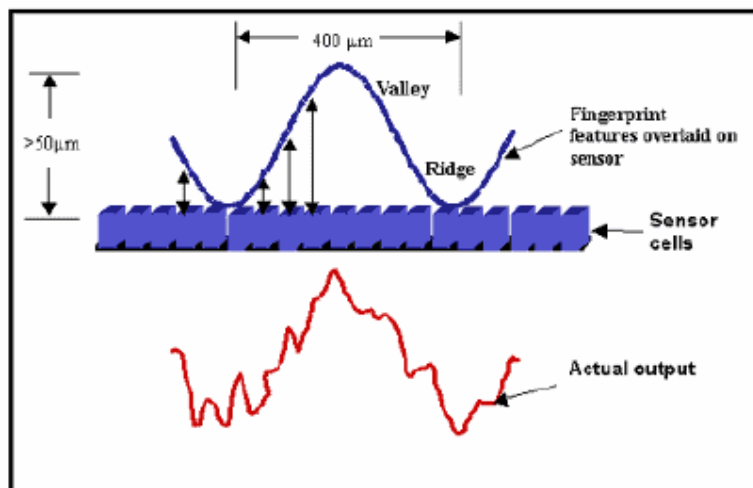


Abbildung 3 Funktionsprinzip des kapazitiven Sensors³

3.3.2 Processing Element

Der Processing Block, bestehend aus dem Image Pre-Processing und dem Matching-Algorithmus, stellt das Gehirn des Systems dar. Hier wird durch diverse Algorithmen das von dem Sensor gelieferte Bild verarbeitet und anschliessend als Template gespeichert oder dem Ausgang-Interface mitgeteilt, ob der anliegende Fingerabdruck erkannt wurde.

3.3.2.1 Image Pre-Processing

Dieser Prozess wird als erstes vor dem Abspeichern des eingescannten Fingerabdrucks durchgeführt, um eine gute Bildqualität des Abdruckes, ein sogenanntes Template, zu erhalten. Wie schon erwähnt, erhält die Image Pre-Processing-Einheit vom Sensor ein 8-Bit Graustufen Bild. Uns würde allerdings ein Fingerabdruck mit 1-Bit-Elementen schon genügen. Denn die Grautöne welche zwischen Rille und Tal eines Abdrucks liegen, können als Störuschen gesehen werden. Wesentlich, sind für uns die Rillen, welche die einmalige Struktur eines jeden Fingerabdrucks widerspiegelt.

Diese 8-Bit- zu 1-Bit-Element Transformation kann mit Hilfe eines Filters realisiert werden, was später unter Punkt 4 noch detailliert beschrieben wird. Das nun entstandene Template ist ein wichtiger Punkt für die spätere Weiterverarbeitung. Denn nur durch ein gutes Template kann der Matching-Algorithmus seine Aufgabe erfüllen. Ist das von ihm gelieferte Bild nicht einwandfrei, kann auch der beste Matching-Algorithmus seine Aufgabe erfüllen.

3.3.2.2 Matching-Algorithmus

Der Matching-Algorithmus hat eine schwere Aufgabe zu bewältigen, da ein Scan eines Fingerabdrucks niemals dem Anderen exakt gleicht, auch wenn es sich immer um den selben Finger handelt. Er muss in der Lage sein, bei unterschiedlicher Positionierung des Fingers, also Translation, Rotation wie aber auch Verformungen durch unterschiedlichem Auflagedruck, den Fingerabdruck zu vergleichen.

Zwei Methoden kommen derzeit meistens zur Anwendung:

a) Die Mustererkennung:

Die Mustererkennung benötigt lediglich ein 1-Bit monochromes Bild des Fingerabdrucks und legt geometrisch strukturierte Formen oder Vermessungen über den Fingerabdruck. Diese Methode hat den Vorteil, wenig Rechenleistung zu

benötigen und wird in Systemen verwendet, in denen Scanner und Auswertung in einer Einheit verbunden sind.

b) Die minutien-basierende Erkennung⁴:

Für die minutien-basierende Erkennung wird zumeist die Grauwertdynamik eines 8-Bit Gray Scale Bildes benötigt. Aus diesen werden die Linienmuster mit seinen Verzweigungen, Inseln und Bruchstellen (Minutien=Details) hervorgehoben und Anfang und Ende, Richtung und Lage markiert. Nun kann man ein Template erzeugen das nur noch aus einer gewissen Anzahl Minutien besteht. Bei dem Erkennungs-Algorithmus muss man daher nur noch die Minutien miteinander vergleichen, um eine Identifikation zu erhalten. In der Abbildung 4 handelt es sich um ein Fingerabdruck mit 32 erzeugten Minutien.



Abbildung 4 Fingerabdruck mit 32 Minutien

Da diese Methode aber durch aufwendige Grafikverarbeitung eher rechenintensiv ist, wird sie nur im Zusammenhang mit leistungsfähigen CPU's oder DSP's eingesetzt.

Wie gesagt sind die oben erwähnten Methoden am Meisten verbreitet. Allerdings ist die Mustererkennung wie auch die minutien-basierende Erkennung sehr umfangreich und insbesondere durch ihre Komplexität schwierig, eine Echtzeimplementation zu realisieren.

Infolgedessen entschieden wir uns für eine dritte Variante, in der wir die zwei zu vergleichende Bilder voneinander subtrahieren. Ist die Differenz so einer Subtraktion klein könnte eine Übereinstimmung vorhanden sein, ist die Differenz gross, ist mit grosser Wahrscheinlichkeit keine Übereinstimmung gegenwärtig. Die ersten Implementationen dieser Variante haben wir auf Matlab bewerkstelligt und wird demgemäss unter Punkt 4 noch detaillierter beschrieben.

⁴ Springer Verlag 2003 minutiae-based-methods.pdf

Unabhängig von der Erkennungsmethode werden zwei Prüfverfahren unterschieden:

a) die Verifikation („one-to-one matching“ - 1:1).

Es ist das derzeit am häufigsten verwendete Verfahren. Dabei werden die extrahierten Daten des zu prüfenden Fingerabdrucks nur mit einem gespeicherten Fingerabdruck verglichen.

In der Praxis bedeutet das, dass bekannt sein muss, welche Daten verglichen werden sollen. Sie müssen also vorangehend mittels PIN oder UserID aufgerufen werden.

b) die Identifikation (“one-to-many matching“ - 1:N).

Dabei werden die extrahierten Daten des zu prüfenden Fingerabdrucks mit allen gespeicherten Fingerabdruck-Daten aus einer Datenbank verglichen. Dafür sind weitaus leistungsfähigere und genauere Prüfverfahren und eine hohe Rechenleistung notwendig, um akzeptable Matchingzeiten zu erreichen.

Da die Realisierung so einer Fingerprint Authentication unser Kapazitäten nicht überschreiten sollte, entschieden wir uns einfach halber für die Verifikation (1:1).

3.3.3 Speicher Element

Das Speichermedium kann eine Festplatte oder im unserem Fall die DSK-Chipkarte sein. Im Speicher werden die durch den Processing Block entstandenen Templates abgespeichert.

3.3.4 Interface Element

Schlussendlich ist das Ausgang-Interface Element, welches die Entscheidung einer Zugangsbewilligung kommuniziert. Dies bewerkstelligen wir anhand zweier LED auf der Fingerprint Daughter-Card. Eine grüne LED zeigt die Übereinstimmung zweier Fingerabdrücke an und die rote LED offenbart, dass keine Deckung der Abdrücke besteht.

Auch das Code Composer Studio, welches später noch in der Entwicklungsumgebung und Echtzeimplementierung genauer beschrieben wird, benützen wir als Benutzeroberfläche um einerseits das mehrmalige Einlesen eines Abdruckes und andererseits das Matching durchzuführen.

3.4 Genauigkeit von biometrischen Methoden

Für die Bewertung eines Fingerabdruck-Erkennungs-System sind folgende Kenngrößen wichtig.

FAR - Die Falsche Akzeptanz Rate bezeichnet die durchschnittliche Häufigkeit, mit der ein anderer Fingerabdruck erkannt wird (Falsche Erkennung, typisch kleiner als 0.001%).

FRR - Die Falsche Reaktion Rate bezeichnet die durchschnittliche Häufigkeit, mit der ein Fingerabdruck nicht erkannt wird (Falsche Ablehnung, typisch kleiner als 0.1%).

EER - Equal Error Rate setzt sich aus FAR und FRR (als Kurve dargestellt) zusammen und beschreibt den Punkt an dem sich beide überschneiden.

Es gilt daher: Je höher die Genauigkeit (FAR), desto grösser die Gefahr einer falsche Ablehnung (FRR) wobei die Wahrscheinlichkeit, einen berechtigten Anwender abzuweisen,

ebenso hoch ist, wie die, eine unberechtigte Person zuzulassen.

In der Abbildung 5 sieht man nun so eine EER-Kurve.

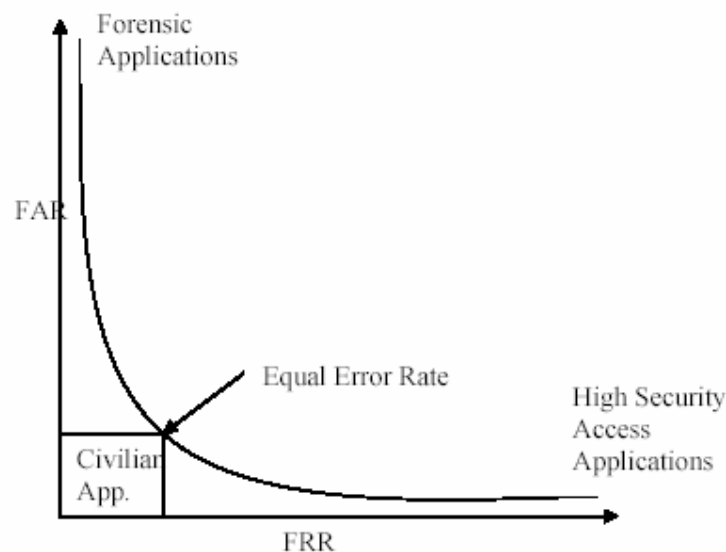


Abbildung 5 Equal Error Rate⁵

Bei der Anwendung eines Fingerabdruck-Erkennungs-Systems steht es dem Betreiber frei die Parameter je nach Anwendungszweck zu bestimmen. Wäre zum Beispiel das Einsatzgebiet in einer Bank um in einen Tresorraum zu gelangen, ist es sinnvoll, wenn der Betreiber die FAR sehr tief setzt, was allerdings eine öfter falsche Ablehnung zur Folge hat, aber die Sicherheit die hier vorrangig ist gewährleistet.

⁵ Michigan State University Biometrics%20(JainRoss).pdf

Wenn hingegen der Einsatzort die HSR wäre, ist es nicht sinnvoll den FAR zu tief zu setzen, da es sonst bei einzelnen Studenten zu Wutausbrüche führen könnten, wenn sie abermals abgelehnt werden.

4. Lösungsvarianten in Matlab

4.1 Über Matlab

Matlab ist ein numerisches Rechenprogramm, welches einfach zu handhaben ist. Es erlaubt mathematische Anwendungen mittels einfachen Befehlen auszudrücken und grafisch darzustellen.

Ebenfalls erlaubt es Bilder in unkomprimierter Form einzulesen (z.B. Datentypen wie tif oder bmp) und zu bearbeiten.

Da wir am Anfang unserer Semesterarbeit noch nicht über die Hardware und dazu mitgelieferten Software (Code Composer Studio) verfügten, war es sinnvoll erste Ideen der Pre-Processing Filter, wie auch Matching-Algorithmus in dieser anwenderfreundlichen Programmiersprache zu implementieren.

4.2 Digitale Bilddarstellung

Unsere eingescannten Fingerabdrücke werden in Matlab als RGB-Bilder angezeigt. In Matlab wird so ein Fingerabdruck mit einer 2-dimensionalen Bildmatrix dargestellt. Handelt es sich bei diesem Bild um ein Graustufen-Bild, wie in unserem Falle, so ist diese Matrix sehr übersichtlich und intuitiv



Abbildung 6 Graustufen-Fingerabdruck

Man kann nun gedanklich in der Abbildung 6 einen feinen Raster über den von dem Sensor gelieferten 8-Bit Fingerabdrucks legen. Bei jedem Kästchen befindet sich nun einen gewissen Helligkeitswert, welcher dann in einer 2-dimensionalen Matrix abgelegt und als Pixel (Picture Element) bezeichnet wird. In der Tabelle 1 ist einen kleinen Ausschnitt so einer Matrix dargestellt. Die Werte müssen bei einem 8-Bit Bild zwischen 0 und 256 liegen.

26	18	10	17	17	17	21	22	28	23	20	17	18	19
25	21	16	18	17	17	12	20	20	27	20	15	18	14
20	22	19	10	18	18	18	17	19	23	27	25	29	26
20	28	25	18	18	18	18	17	18	20	21	25	28	24
22	20	25	21	18	10	17	12	15	16	18	24	29	27
12	25	21	27	22	18	10	17	14	17	17	15	12	26
14	26	28	20	25	20	18	18	18	17	10	10	17	10
17	21	22	20	29	22	21	17	10	16	18	10	18	18
10	17	27	23	20	28	23	18	18	19	21	20	17	17
18	16	16	27	24	29	29	18	10	17	21	29	28	18
17	14	17	17	26	25	23	19	19	18	14	21	27	20
18	18	18	17	10	23	25	29	10	17	19	14	16	24
21	18	17	18	18	16	22	23	23	18	19	18	18	17
24	24	15	18	18	17	24	21	22	20	18	12	17	18
27	20	17	16	19	18	14	12	29	22	29	20	18	17

Tabelle 1 Kleiner Ausschnitt eines Fingerabdruckes in Matrix Darstellung

Es liegt auf der Hand, dass die Qualität eines Fingerabdruck-Bildes von der Anzahl der Pixel abhängt, sowie der Unterteilung der Helligkeitsstufen, welches Beides, bei besserer Qualität, mit grösserem Speicherplatz verbunden ist.

In unserem Fall liefert unser FPC1010 Sensor ein 200x152 Pixel-Bild mit 8 Bits (1 Byte) pro Pixel für die Helligkeitsunterscheidung, so sind $2^8 = 256$ Helligkeits-Abstufungen möglich. In Matlab ist die Darstellung dann so, dass in jedem Matrixelement (Pixel) ein Wert zwischen 0 und 255 steht. Dabei entspricht der Wert 0 schwarz und der Wert 255 weiss. Alle Werte dazwischen sind Graustufen. Auch wird ersichtlich wie viel Speicherplatz so ein Fingerabdruck-Bild braucht: $200 \times 152 = 30400$ Pixels à 1 Byte was 30.4KBytes entspricht.

4.3 Pre-Prozessing

Wie schon erwähnt, wird im Pre-Prozessing durch Filterung aus einem 8-Bit Graubild ein schwarz-weiss Template erzeugt, welches dann vom Matching Block benützt wird, um ein Gespeichertes und ein anliegendes Template miteinander zu vergleichen. Für dieses Template wären daher Pixels mit 1-Bit anstatt mit 8 ausreichend.

Dadurch wird jedem Fingerabdruck ein Template zugeordnet. Dies allerdings kann zu Problemen führen, da der Finger nicht immer genau gleich auf den Sensor platziert wird. Deshalb entstehen für den gleichen Fingerabdruck, verschiedene Templates. Die Translation nach oben, unten, links, rechts ist dabei weniger das Problem, dies kann im Matching-Algorithmus elegant gelöst werden.

Allerdings die Rotation und der Druck sind eher die Knacknuss, die nicht mehr so einfach im Matching zu bewältigen sind.

So werden wir diese Knacknuss in unserem Pre-Prozessing lösen. Die Idee ist ganz einfach. Wir erzeugen anstatt nur 1 Template, mehrere Templates die einen Fingerabdruck repräsentieren.

Ein Fingerprint wird genau dreimal eingelesen wie in der Abbildung 7 zu sehen ist. Einer Gerade, einer leicht nach links gedreht und einer leicht nach rechts gedreht. Dies einmal mit starkem Auflagedruck und einmal mit leichtem. Daraus entstehen genau sechs Templates für einen Fingerabdruck. Stärkere Rotationen werden wir nicht berücksichtigen, da durch die physikalische Form des Sensors die Auflageposition des Fingers ziemlich genau gegeben ist.



Abbildung 7 Auflageposition der 3 Templates

4.3.1 Filter

Das Prinzip dieses Filters ist einfach. Zuerst wird der durchschnittliche Graustufenwert berechnet. Dieser Wert wird als Schwellwert benutzt. Später wird dann jeder Pixel mit dem Schwellwert verglichen. Wenn der Wert unter dem Schwellwert liegt wird er auf null (schwarz) gesetzt, wenn er darüber liegt auf 255 (weiss). So haben wir nur noch ein schwarz-weiss Bild, das auch weniger Störeinflüsse hat.

$$mean = \frac{\sum_{i=0}^c \sum_{j=0}^d Pixel(i, j)}{c * d}$$

Formel 1 Berechnung des durchschnittlichen Graustufenwertes

Wobei der mean der durchschnittliche Graustufenwert eines Fingerabdrucks entspricht, c die Zeilen- und d die Spaltenanzahl.

4.3.2 Filterversuch

Fingerabdruck vor dem Filter:

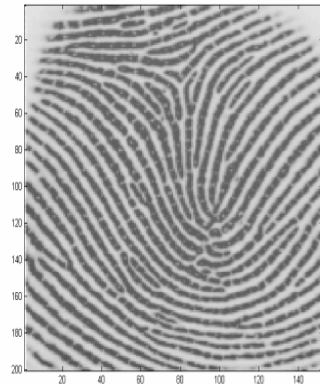
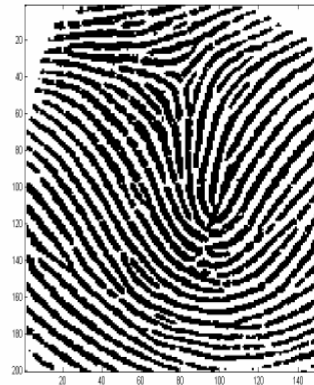


Abbildung 8 Fingerabdruck vor Filter

Fingerabdruck nach dem Filter:



Fingerabdruck nach Filter

Abbildung 8 stellt einen Fingerabdruck vor und nach dem Filter dar. Nach dem der Fingerabdruck gefiltert wurde, entsteht ein Template welches ersichtlich nur noch aus Schwarz- und Weiss-Anteilen besteht.

4.3.3 Der Matlab Code des Filters:

```
function [mask] = df(A)

    summenVector=sum(A);
    summe=sum(summenVector);
    [c,d]=size(A);
    durchschnitt=summe/(c*d);

    for i=1:c;
        for j=1:d;
            if A(i,j) > durchschnitt
                A(i,j) = 255;
            else
                A(i,j) = 0;
            end;
        end;
    end;

    mask = A;
```

Bei Beginn der Implementierung verwendeten wir Fingerabdrücke von dem Internet, welche qualitativ schlecht waren. Aus diesem Grunde konnten wir nicht den durchschnittliche Graustufenwert so einfach, wie im Matlab-Code oben dargestellt, berechnen. Glücklicherweise stellte sich heraus, dass unseren Sensor FPC1010 qualitativ viel bessere Bilder lieferte, welche uns ermöglichte, einen einfachen Filter dafür zu benutzen.

4.3.4 Darstellung des vom Sensor gelieferten Fingerabdrucks auf Matlab

Nun war es an der Zeit, die vom Sensor erhaltene Fingerabdrücke mit unseren auf Matlab entwickelte Pre-Prozessing- und Matching-Algorithmen aus zu testen. Allerdings schafften wir es nicht auf Anhieb, das Bild von dem CCS in Matlab zu bringen.

Es stellte sich heraus, dass es leider nicht so einfach ist, wie wir anfangs dachten. Das Problem bestand darin, dass man im CCS nicht einfach Bilder abspeichern kann, sondern nur den Speicher an der Adresse des Bildes auslesen kann und die Daten in verschiedenen Zahlenformaten abspeichern kann. Wer schon mal CT-Unterricht genossen hat, kennt sicher das Little-Endian und Big-Endian Problem beim Speicher auslesen. Dieses Problem hat uns auch einige Zeit beschäftigt.

Methode um aus HEX-Zahlen Bilder zu machen

```
function [bild] = machBild(m)

    [a b c d e] = textread(m,'%2c%2c%2c%2c%2c');

    B=hex2dec(b);
    C=hex2dec(c);
    D=hex2dec(d);
    E=hex2dec(e);
    x=0;
    for i=1:200
        for j=1:38
            x=x+1;
            bild(i,(j-1)*4+1) = E(x);
            bild(i,(j-1)*4+2) = D(x);
            bild(i,(j-1)*4+3) = C(x);
            bild(i,(j-1)*4+4) = B(x);
        end;
    end;

    figure(1); image(bild);cm=[0:1/254:1;0:1/254:1;0:1/254:1];colormap(cm);
```

Der oben stehende Code macht aus einer Datei gefüllt mit 8-stelligen HEX-Zahlen ein darstellbares Bild.

4.4 Matching-Algorithmus

4.4.1 Funktion

Der Matching-Algorithmus schiebt lediglich das Bild des anliegenden Fingerabdrucks um 1/8 der Länge und Breite über das abgespeicherte Template und berechnet relativ zur gedeckten Fläche die Differenz aus. Diese Differenz wird dann mit einem empirisch bestimmten Schwellwert verglichen. Liegt die Differenz unter diesem Schwellwert besteht eine Übereinstimmung der zwei verglichenen Fingerabdrücke vor.

$$relativeDifferenz = \frac{\sum_{i=0}^c \sum_{j=0}^d |Bild1(i, j) - Bild2(i, j)|}{c * d}$$

Formel 2 Berechnung der relativen Differenz

4.4.2 Matching 1

Es wurde zweimal der gleiche Fingerabdruck (Abbildung 9) nacheinander eingelesen und miteinander verglichen

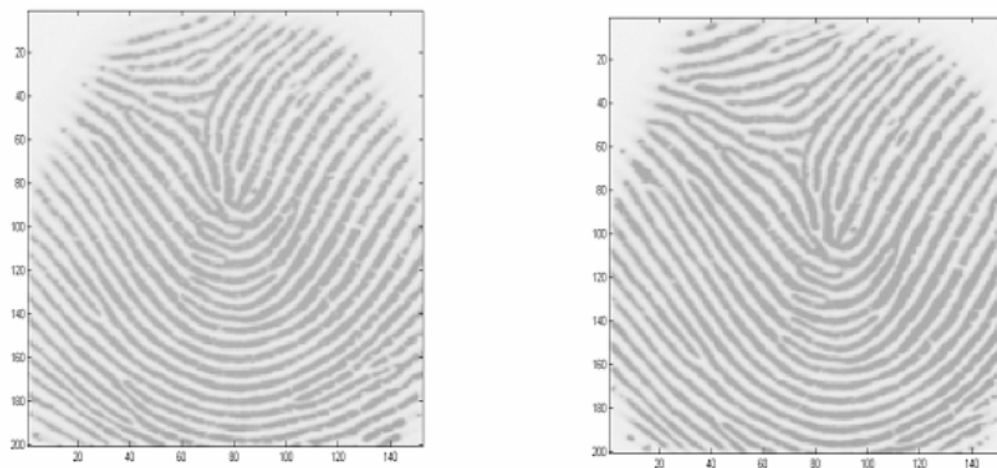


Abbildung 9 Zwei gleiche Fingerabdrücke

Die relative Differenz der beiden Abdrücke beträgt 15,42 pro Pixel

4.4.3 Matching 2

Nun wurden zwei verschiedene Fingerabdrücke (Abbildung 11) miteinander verglichen

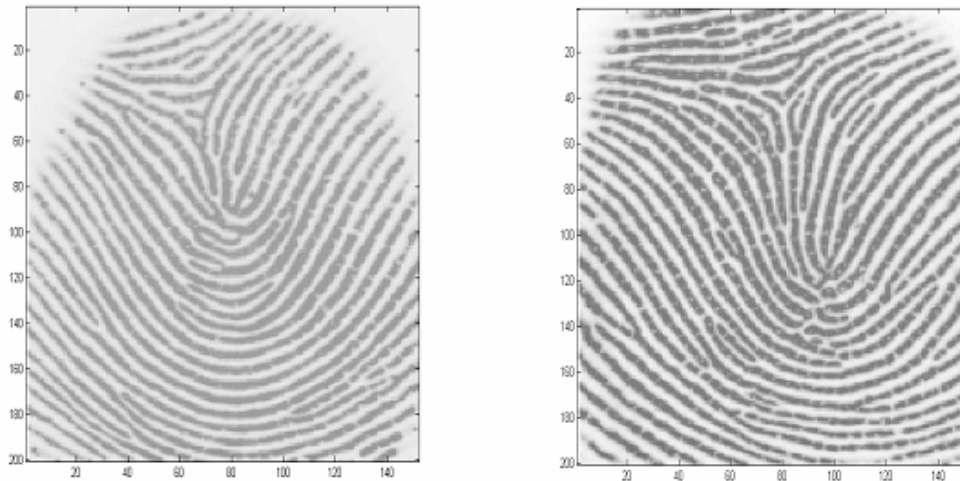


Abbildung 10 Zwei unterschiedliche Fingerabdrücke

Die hier entstandene relative Differenz beträgt nun 22,4 pro Pixel.

4.4.4 Schlussfolgerung

Wie im Versuch 1 und Versuch 2 ersichtlich liegt die relative Differenz im Versuch 1 tiefer als im 2. Versuch. Dies war auch vorhersehbar, da es sich bei 1. um den gleichen Fingerabdruck handelte. Dennoch ist eine Abweichung bei Versuch 1 vorhanden, was wiederum verständlich ist, da so zu sagen niemals der Finger zweimal exakt gleich auf den Sensor platziert wird.

Hier wurde der anliegende Fingerabdruck nur mit einem Template verglichen, da wir hier in dieser Phase lediglich den Matching-Algorithmus testen wollten. Aus diesem Grund haben wir darauf geachtet den Finger zweimal ohne Rotation und verschiedenem Druck auf dem Sensor zu legen.

Jedoch ist immer noch nicht bekannt wie gut eigentlich die relative Differenz bei dem Versuch 1 ist. Wie schon gesagt, wird eine Übereinstimmung zweier Abdrücke durch den relativen Schwellwert bestimmt.

4.4.5 Schwellwert

Den Schwellwert lässt sich nicht so einfach durch eine mathematische Formel bestimmen. Er muss empirisch bestimmt werden. Deshalb haben wir dieses Matching abermals mit verschiedenen Fingerabdrücke durchgespielt, um zu sehen ob unsere erhaltenen relativen Differenzen oben nicht nur durch Zufall so ausgefallen sind.

Relative Differenz von zwei gleichen Fingerabdrücke:

Halter des Fingerprints	Baumann	Kropaci	Plaz	Lang
Relative Differenz	14,83	16,76	14,30	15,42
	17,32	13,97	15,83	16,07

Relative Differenz von zwei unterschiedlichen Fingerabdrücke:

Halter des Fingerprints	Baumann,Lang	Kropaci,Lang	Plaz,Lang	Plaz, Baumann
Relative Differenz	23,57	24,41	23,25	23,87

Halter des Fingerprints	Kropaci,Plaz	Kropaci,Baumann		
Relative Differenz	24,97	25,34		

Hier wurde unser vorhergehendes Ergebnis nochmals bestätigt. Grob könnte man sagen, dass wenn die relative Differenz unter 20 liegt besteht eine Übereinstimmung und, falls es unter der 20er Marke liegt, kann man von zwei gleichen Abdrücken ausgehen. Uns ist allerdings klar, dass diese wenige Versuche nicht genügen, um einen genauen Schwellwert zu bestimmen. Dies war allerdings auch nicht unsere Absicht in diesem Abschnitt unserer Semesterarbeit. Wir werden in einer späteren Phase den Schwellwert durch weitere Versuche bestätigen oder neu bestimmen.

4.4.6 Matlab-Code des Matching-Algorithmus:

A,B: Fingerabdrücke
r: Zeile
c: Spalte
i,j,k: Laufvariablen

```
function [minDifferenz] = match(A,B)

[r,c] = size(A);
minDifferenzxx = 99999;

for i=1:r/8
    for j=1:c/4

        differenz(1) = (sum(sum(abs(A(1:(r-i+1),1:(c-j+1))-B(i:r,j:c)))))/((r-i+1)*(c-j+1));
        differenz(2) = (sum(sum(abs(A(i:r,j:c)-B(1:(r-i+1),1:(c-j+1)))))/((r-i+1)*(c-j+1));
        differenz(3) = (sum(sum(abs(A(1:(r-i+1),1:(c-j+1))-B(1:(r-i+1),1:(c-j+1)))))/((r-i+1)*(c-j+1));
        differenz(4) = (sum(sum(abs(A(i:r,j:c)-B(i:r,j:c)))))/((r-i+1)*(c-j+1));

        for k=1:4
            if(differenz(k) < minDifferenzxx)
                minDifferenzxx = differenz(k);
                differenz(k) = 0;
            end;
        end;
    end;
end;

minDifferenz = minDifferenzxx;
```

5 Entwicklungsumgebung

Nachdem wir das Fingerprint Authentication Development Tool erhielten, machten wir uns als erstes mit der Entwicklungsumgebung ein wenig vertraut, bevor wir mit der Inbetriebnahme der Daughter-Card begannen.

Bei der Entwicklungsumgebung handelt es sich um 3 Hauptkomponenten.

5.1 TMS320CDSK6711 DSK

Eckdaten des Entwicklungsbords:

- CPU: 150-MHz C6711DSP, mit der Leistung von 900 Millionen Floating Point Rechenoperationen pro Sekunde(MFLOPS)
- 16M Bytes Synchronous Dynamic Random Access Memory (SDRAM)
- 64K Bytes internen Speicher L2
- 128K Bytes Flash
- Enhanced direct-memory-access (EDMA) controller
- Two multi-channel buffered serial ports (McBSPs)

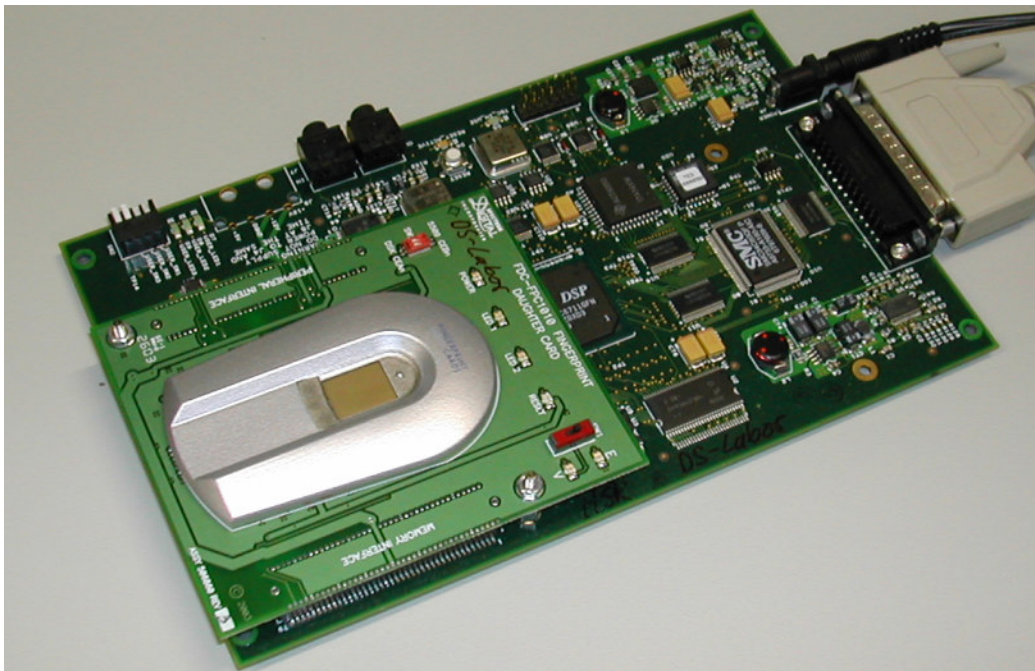


Abbildung 11 6711DSK-Board mit Daughter-Card

Um Daten von der Tochterkarte auf das DSK-Board zu senden, braucht es noch eine kleine Modifikation. Man muss den Pin 1 und 2 auf dem Jumper 1 verbinden.

Jumper #	Position
JP1	1-2

5.2 FPC1010 Finger Print Sensor Daughter Card

Bei der Tochterkarte handelt es sich hauptsächlich um einen Fingerprint-Sensor. Die Karte wurde speziell für den Gebrauch mit der DSK oder dem EVM entwickelt. Man benutzt einen kapazitiven Sensor mit einem SPI-Typ Interface.

Die Schnittstelle zwischen der DSK-Karte und der Tochterkarte läuft via McBSP, ersichtlich anhand der Abbildung 12.

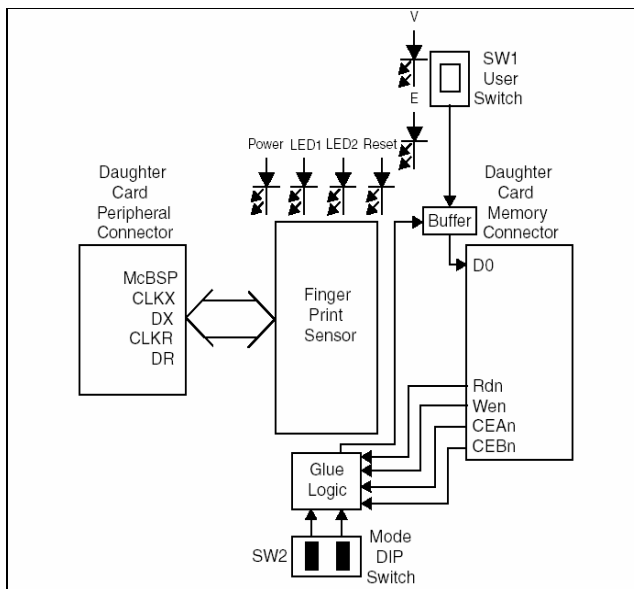


Abbildung 12 Blockscha der Tochterkarte⁶

⁶ Technische Referenzen des FPC1010 FPC1010_Daughtercard_TechRef.pdf

Konfigurationen der Schalterstellung⁶:

SW2-2	McBSP Routing
Off	5509 EVM McBSP Select
On	DSK McBSP Select

Eckdaten der FPC1010 Fingerprint Sensor Daughter Card:

- 200 x 152 Pixel kapazitiver Fingerprint Sensor
- 3.3 volt operation
- Mode selection switch
- User selection switch
- Sechs Status LEDs

5.3 Code Composer Studio

Die Entwicklungsumgebungssoftware Code Composer Studio v2 (CCS) ist einer der Schlüssel-Komponenten, für die Programmierung des DSP auf dem DSK6711-Bord. Sie eignet sich hervorragend für eingebetete Echtzeit Applikationen, wie in unserem Fall.

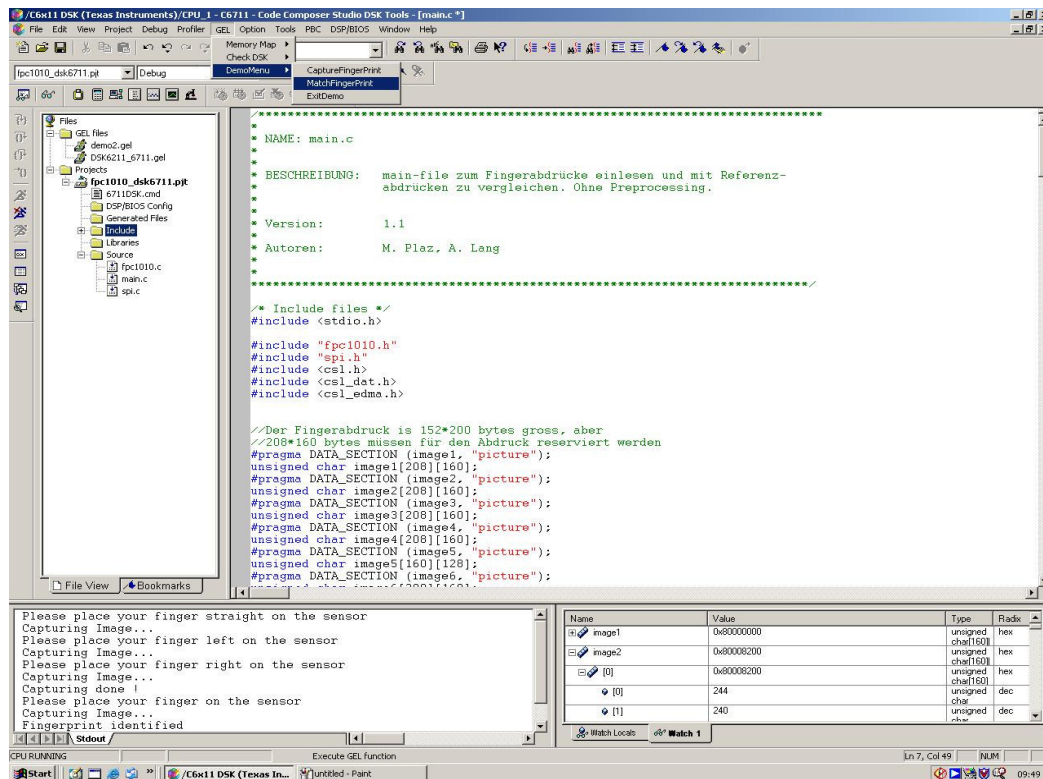


Abbildung 13 Code Composer Studio

CCStudio Vorteile:

- Schneller Start mit bekannten Tools und Benutzeroberflächen
- Einfaches Handhaben von grossen Multi-Prozessor Projekten.
- Einfaches kreieren von Codes durch Optimierungs- und Debugging-Tools
- Multimedia "help-files", und ausführliche Dokumentation für einen schnellen Start
- Spielt komplexe DSP Simulationen in Minuten statt Stunden ab

CCStudio Eigenschaften:

- Integrierte Entwicklungsumgebung mit eng integriertem Editor, Debugger und visuelles Projekt-Management System
- Fortgeschrittenes “**Watch-Fenster**, um lokale und globale Variablen anzuschauen
- **Real-time analysis and data visualization tools**
- **GEL Funktion**
- C/C++ Kompiler, Assembler und Linker (code generation tools)
- Real-Time Data Exchange (RTDX) zwischen host and target
- C++, UML, MATLAB und VAB Unterstützung im Kompiler
- File I/O Probe Points and graphical algorithm scope probes
- Interactive profiling and advanced graphical signal analysis
- Multi-Prozessor Debugging

6 Inbetriebnahme der Daughter Card

Das Fingerprint Authentication Development Tool lieferte uns die Treiber-Software die zum Ansteuern der Daughter-Card notwendig war und ein C-Programm zum Einlesen von Fingerabdrücken mit den dazugehörenden Header Dateien.

Wir haben die Software mit dem Code Composer Studio auf den DSP geladen und versuchten unsere ersten Fingerabdrücke einzulesen. Doch als wir erstmals das Graphical Display Window des Code Composer Studio betrachteten, sahen wir alles andere als ein Fingerabdruck. Nach erneutem durcharbeiten der mitgelieferten Unterlagen fanden wir heraus das wir die Einstellungen des Graphical Display Window zu ändern haben. So änderten wir die Einstellung wie unten ersichtlich.

Graph Property Dialog Box Settings:

Color space:	RGB
Interleaved data:	Yes
Start address:	image (Image Array definiert im Kode)
Lines per display:	200
Pixels per line:	152
Byte packing to fill 32 bits:	NO
Bits per pixel:	8
Palette Option:	Gray scale of 256 colors
Image Origin:	Top Left
Everything else:	Default

Aber auch dies hatte keine Verbesserung des dargestellten Bildes ergeben. Es war immer noch alles andere als ein Fingerabdruck zu erkennen.

Schlussendlich fanden wir heraus, dass man das DSK 6711 Board für unsere benutzte Daughter-Card speziell zu konfigurieren hat. Man muss den Pin 1 und 2 beim Jumper JP1 verbinden.

Und nachdem wir durch einen Jumper eine Verbindung der Pins erzeugten, erhielten wir zu guter Letzt einen schönen Fingerabdruck.

7 Echtzeitimplementierung

Schlussendlich war es an der Zeit, die im Matlab erzeugten Algorithmen mit dem DSP-Prozessor in Echtzeit zu realisieren. Bevor wir aber mit den Programmieren der Algorithmen beginnen konnten, mussten wir zuerst einen Überblick über die Speicherbelegung erarbeiten, um festzustellen, wo überhaupt unsere Daten der Fingerabdrücke liegen und wie wir auf sie zugreifen, um ein System in Echtzeit zu entwickeln. Außerdem mussten wir das Problem angehen, wie wir eigentlich das Einlesen oder Vergleichen von Fingerabdrücke auslösen. Doch dieses Problem liess sich sehr elegant mit Hilfe des GEL lösen.

7.1 Speicherbelegung des 6711 DSK

Die Tabelle 2 zeigt den Adressbereich der DSK6711-Karte. Interne Speicher ist immer ab der Adresse 0x0 und kann als Programm- oder Daten-Speicher benutzt werden. Weiter befinden sich die Konfigurations-Register der Peripherie-Komponenten in diesem Bereich.

Die externen Speicher existierten ab der Adresse 0x8000 0000.

Neben den Registern zur Datentransfer, benützen wir vor allem den internen Speicher (L2) und den Externen (SDRAM). Beim L2 handelt es sich um einen 64K Bytes grossen Speicher der einen sehr schnellen Zugriff erlaubt, da er mit dem CPU-Clock (150MHz) getaktet wird. Der SDRAM (16 MBytes) hingegen wird mit 100MHz getaktet. Dies bedeutet im Gegensatz zum L2 eine 1/3 langsamere Zugriffszeit.

MEMORY BLOCK DESCRIPTION	BLOCK SIZE (BYTES)	HEX ADDRESS RANGE
Internal RAM (L2)	64K	0000 0000 – 0000 FFFF
Reserved	24M – 64K	0001 0000 – 017F FFFF
External Memory Interface (EMIF) Registers	256K	0180 0000 – 0183 FFFF
L2 Registers	256K	0184 0000 – 0187 FFFF
HPI Registers	256K	0188 0000 – 018B FFFF
McBSP 0 Registers	256K	018C 0000 – 018F FFFF
McBSP 1 Registers	256K	0190 0000 – 0193 FFFF
Timer 0 Registers	256K	0194 0000 – 0197 FFFF
Timer 1 Registers	256K	0198 0000 – 019B FFFF
Interrupt Selector Registers	256K	019C 0000 – 019F FFFF
EDMA RAM and EDMA Registers	256K	01A0 0000 – 01A3 FFFF
Reserved	6M – 256K	01A4 0000 – 01FF FFFF
QDMA Registers	52	0200 0000 – 0200 0033
Reserved	736M – 52	0200 0034 – 2FFF FFFF
McBSP 0/1 Data	256M	3000 0000 – 3FFF FFFF
Reserved	1G	4000 0000 – 7FFF FFFF
EMIF CE0†	256M	8000 0000 – 8FFF FFFF
EMIF CE1†	256M	9000 0000 – 9FFF FFFF
EMIF CE2†	256M	A000 0000 – AFFF FFFF
EMIF CE3†	256M	B000 0000 – BFFF FFFF
Reserved	1G	C000 0000 – FFFF FFFF

Tabelle 2 Memory Mapping⁷

⁷ Texas Instrument spru197d.pdf

7.2 GEL (General Extension Language)

Die GEL ist eine interpretierende Sprache, die C ähnlich ist. Mit ihr kann man die Benutzerfreundlichkeit im Code Composer Studio ausbauen. Man kann es als GUI betrachten, da man Funktionen über das GEL Menu aufrufen kann.

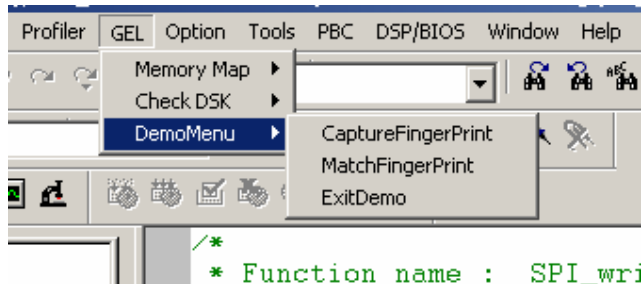


Abbildung 14 GEL-Menu

In unserer Arbeit benutzen wir diese Eigenschaft des GEL's um unsere Matching- und Capture-Fingerprint Funktionen zu starten. Durch das Auslösen der GEL-Funktion, wird lediglich eine globale boolean Variable auf eins gesetzt, die in der main-Funktion eine IF-Bedingung erfüllt und durch goto zur richtigen Position springt.

Im demo.gel

```
hotmenu CaptureFingerPrint()
{
captureFP = 1;

GEL_Go(dummyFunc1);
GEL_Run();
}
```

Im main.c

```
if(captureFP){
    goto capture;
}
```

7.3 Datenübertragung

7.3.1 McBSP

Der 6711 DSK verfügt über zwei McBSP Ports (Multi Channel Buffered Serial Port)

McBSP0: An diesem seriellen Port ist der A/D- sowie der D/A-Wandler des DSK angeschlossen.

McBSP1: Dieser serielle Port führt auf den Stecker J3. Beim J3 handelt sich es um einen 80pin-Stecker, über den man eine beliebige Tochterkarte ansprechen kann.

Kurz gesagt handelt es sich um eine Schnittstelle, um einfach auf eine Peripherie mit Hilfe von 32-Bit Registern zuzugreifen.

Wie schon in der Entwicklungsumgebung angedeutet, benützen wir den McBSP, um die Daten des Fingerabdrucksensors auf der Tochterkarte in den externen Speicher (SDRAM) des 6711 DSK zu transferieren.

Die Hauptblöcke des McBSP sind in der Abbildung 15 ersichtlich.

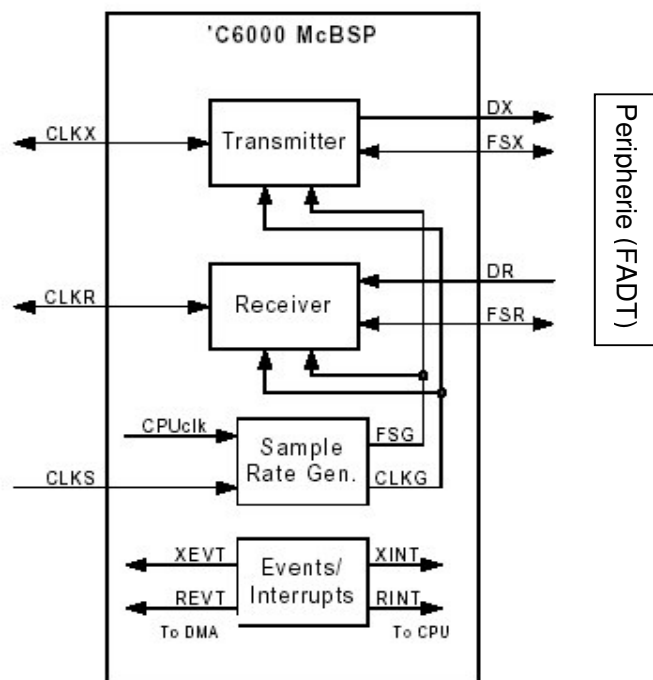


Abbildung 15 Blockscheema des McBSP⁸

⁸ Texas Instrument SPRA488.pdf

Sie bestehen aus:

Transmitter:

Der Übertragungsblock ist für das serielle Senden von Daten verantwortlich, welche sich im DX-Register befinden. Der Inhalt des DXR wird in das Transmit-Shift-Register (XSR) kopiert und sobald ein Transmit-Frame-Sync (FSX) anliegt, bitweise bei jedem Takt (CLKX) ausgegeben. Neue Daten können entweder mit dem CPU oder dem EDMA in das DXR geschrieben werden.

Receiver:

Die Daten die am DR-Pin anliegen, werden taktweise ins Receive Shift Register (RSR) kopiert.

Wiederum werden die Daten vom RSR, bei anliegendem Receive-Fram-Sync (FSR), über das Receive-Buffer-Register (RBR) ins Data-Receive-Register (DRR) geschoben. Das DRR kann durch den CPU oder den EDMA gelesen werden.

Sample Rate Generator:

Hier werden die Kontroll-Signale generiert wie Senden/Empfang Takt und fram-sync Signale, um Daten zu senden und zu empfangen. Den Takt kann man entweder vom CPU (CPUclk) oder von einer externen Quelle (CLKS)wählen.

Events/Interrupt Generation:

Der McBSP generiert sync-Events um dem EDMA mitzuteilen, dass er bereit ist neue Daten zu senden (DRR) oder zu empfangen (DXR). Es gibt read-sync-event (REVT) und write-sync-event (XEVT).

Das gleiche Prinzip mit dem CPU, nur handelt es sich hier um die Interrupts (RINT und XINT).

In der Tabelle 3 ist ersichtlich an welcher Adresse sich die oben genannten Register befinden.

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0190 0000	DRR1	Data receive register via Peripheral Bus	The CPU and DMA/EDMA controller can only read this register; they cannot write to it.
0x3400 0000 – 0x37FF FFFF	DRR1	McBSP1 data receive register via EDMA Bus	
0190 0004	DXR1	McBSP1 data transmit register via Peripheral Bus	
0x3400 0000 – 0x37FF FFFF	DXR1	McBSP1 data transmit register via EDMA Bus	
0190 0008	SPCR1	McBSP1 serial port control register	
0190 000C	RCR1	McBSP1 receive control register	
0190 0010	XCR1	McBSP1 transmit control register	
0190 0014	SRGR1	McBSP1 sample rate generator register	
0190 0018	MCR1	McBSP1 multichannel control register	
0190 001C	RCER1	McBSP1 receive channel enable register	
0190 0020	XCER1	McBSP1 transmit channel enable register	
0190 0024	PCR1	McBSP1 pin control register	
0190 0028 – 0193 FFFF	–	Reserved	

Tabelle 3 Speicherbelegung der McBSP-Register⁹

⁹ Texas Instrument SPRA488.pdf

7.3.1 EDMA

Nun konnten wir ein erstes Mal unseren Matching Algorithmus testen. Doch es stellte sich heraus, dass eine Bearbeitung unserer speichermässig sehr grossen Fingerabdrücke (30K Bytes), die im SDRAM liegen, enorm viel Zeit in Anspruch nahm. Um eine schnellere Abarbeitung zu erzielen, lag es auf der Hand den internen Speicher L2 zu nutzen, da er wesentlich schneller ist. Leider konnten wir nicht alle Daten vom SDRAM in L2 kopieren, da der L2 mit 64K Bytes beschränkt ist. So entstand die Aufgabe die Daten vom SDRAM Blockweise in den L2 kopieren und fortlaufend abzuarbeiten. Dafür eignete sich hervorragend der EDMA-Kontroller

Der EDMA-Kontroller (enhanced direct memory access) kontrolliert alle Daten-Transfers zwischen dem Level-2 Speicher (IRAM, cache) und den Externen auf der C6711 DSK. Es handelt sich um einen sehr schnellen Datentransfer, welcher unabhängig von der CPU geschieht. In der Abbildung 16 ist klar ersichtlich, wie die verschiedenen Kontroller zueinander stehen. In unserem Fall benützen wir den EDMA-Kontroller via EMIF, um unsere Fingerprint-Daten vom externen Speicher (SDRAM) in den internen zu kopieren, um so eine erhöht zeitliche Abarbeitung der Matching-Funktion zu erlangen.

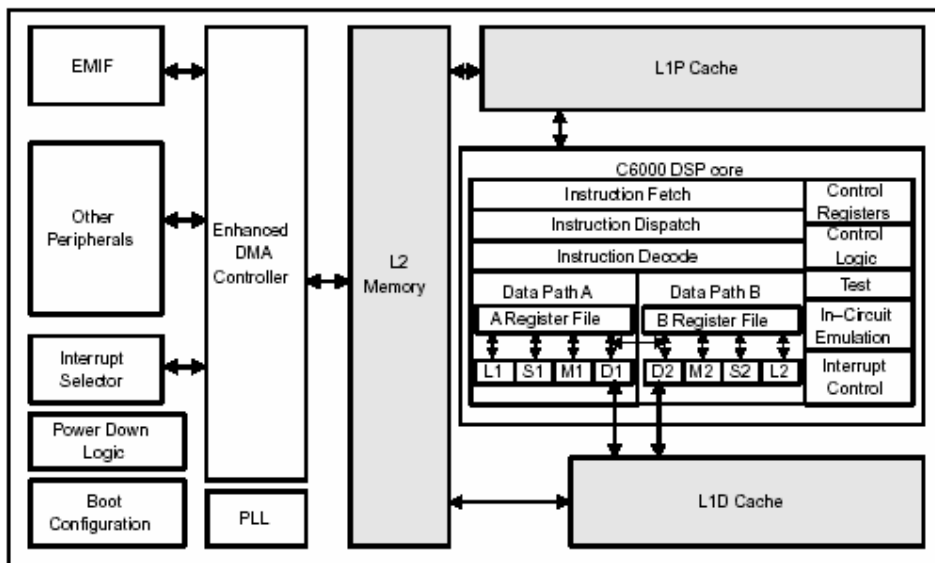


Abbildung 16 Blockdiagramm des DSK6711¹⁰

¹⁰ Texas Instrument spru190d.pdf

Der 6711 verfügt über 16 EDMA-Kanäle. Jeder Kanal, kann unabhängig voneinander initialisiert werden. Weiter beinhaltet der EDMA-Kontroller Event- und Interrupt-Register, Event-Encoder, Parameter RAM, und Address-Generation-Hardware .

EDMA-Events befinden sich im Event-Register, und es handelt sich um ein Synchronisations-Signal, welches ein EDMA-Kanal verursacht, ein Transfer zu starten.

Der EDMA-Kontroller hat auch die Fähigkeit einmalige Transfers zu bewältigen mit Hilfe des QDMA (quick direct memory access). In diesem Fall braucht es auch keine aufwendige Events für die aufwendige Synchronisation wie bei länger andauernden Datenflüssen (Audio/Video).

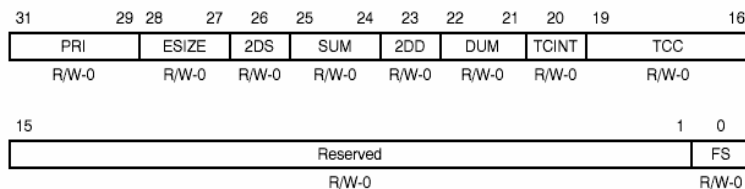
Wir entschieden uns daher, für einen Blocktransfer mit QDMA, da wir nur einmalige Datentransfers unserer Fingerabdrücke haben.

Zur Initialisierung des QDMA sind die folgenden fünf Register à 32-Bit von Bedeutung.

Address		QDMA register
0200 0000h	QDMA Channel Options	QOPT
0200 0004h	QDMA Channel Source Address (SRC)	QSRC
0200 0008h	Array/frame count (FRMCNT) Element count (ELECNT)	QCNT
0200 000Ch	QDMA Channel Destination Address (DST)	QDST
0200 0010h	Array/frame index (FRMIDX) Element index (ELEIDX)	QIDX

Tabelle 4 Register der QDMA¹¹

Unter QDMA Channel Options (QOPT) können folgende Einstellungen zu den DMA Kanälen festgelegt werden:



Legend: R/W= Read/Write; -n = value after reset

Note: QOPT is read/writable; QSOPT is write-only.

Tabelle 5 Channel Option Register des QDMA¹²

¹¹ Texas Instrument spru234.pdf

¹² Texas Instrument spru234.pdf

- PRI:** Hier wird festgelegt, unter welcher Priorität der Kanal arbeitet.
001: High priority
010 Low priority
- ESIZE:** Hier kann die Grösse des zu übertragenen Elements festgelegt werden
00: 32-Bit Element
01: 16-Bit Element
10: 8- Bit Element
- SUM:** Angabe über die Source
00: Fixe Adresse
01: Adresse erhöhen
10: Adresse verringern
- DUM:** Angabe über die Destination
00: Fixe Adresse
01: Adresse erhöhen
10: Adresse verringern
- TCINT:** Wird dieses Bit gesetzt, so wird ein Interrupt ausgelöst, sobald die Übertragung des DMA abgeschlossen ist.

7.3.3 Emif

Der Emif (External Memory Interface) Kontroller verwaltet den externen Speicher ab der Adresse 0x8000 0000. Er wird benutzt für den Datentransfer zwischen dem externen Speicher auf der Tochterkarte (über Stecker J1), dem SDRAM, dem Flash Rom und dem internen Speicher L2.

Wir benützen den EMIF zusammen mit dem EDMA für den Datentransfer unserer Fingerabdrücke vom SDRAM zum Internen Speicher L2.

Unten in der Abbildung 17 sieht man ein Blockschema wie die zwei Blöcke miteinander verbunden sind.

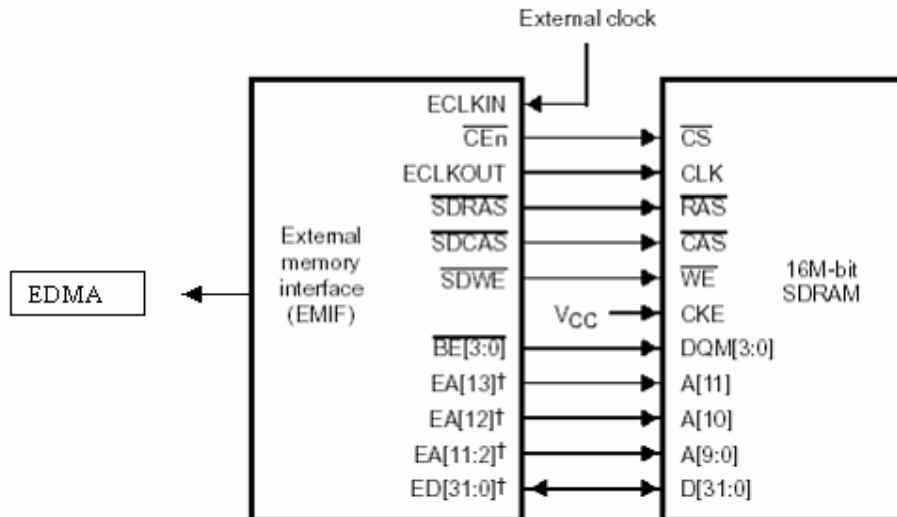


Abbildung 17 Blockschema des EMIF¹³

Beim Initialisieren des EMIFs sind die folgenden Register von Bedeutung:

- SDCTL - SDRAM control register
- SDTIM - SDRAM timing register
- SDEXT - SDRAM extension register

¹³ Texas Instrument spru190d.pdf

8 Schlusswort

Am Anfang unserer Semesterarbeit begannen wir uns über die Grundlagen der Fingerabdruckererkennung zu informieren. Wir verbrachten sehr viel Zeit damit, uns mit dem Pre-Processing und vor allem mit dem Matching auseinander zu setzen. Wir haben uns dabei auf viel komplexe Varianten des Matchings fixiert. Es wurde uns dann in Gesprächen mit Herrn Schuster klar, dass wir uns erstmals auf eine eher einfachere Variante konzentrieren sollten.

Mit unserer Methode haben wir zwar ein schnelles, aber nicht sehr zuverlässiges, System zur Fingerabdruckererkennung geschaffen.

Eine Weiterführung der Arbeit in Richtung Minutiae-Methode scheint uns als zweite Semester- und/oder Diplomarbeit möglich. Diese Methode beschäftigt sich mit markanten Merkmalen, wie Verzweigungen und Enden, des Fingerabdrucks und wird unser Erachtens Zuverlässiger als unsere Methode sein.

Es war eine sehr interessante und lehrreiche Arbeit. Es sind Probleme aufgetaucht an die wir anfangs nicht gedacht hatten. Bei der Entgegennahme der Semesterarbeit stellten wir uns vor, dass wir uns eher mit Bilderkennung und Matching-Algorithmen auseinander setzen würden. Hingegen beschäftigten wir uns sehr lange mit dem DSK6711-Board. Wir befassten uns sehr viel mit dem Aufbau so eines Embedded-Systems. Wir mussten uns mit, für uns, ganz neuen Problemen auseinandersetzen und lernten dadurch viel über verschiedene Möglichkeiten von Datenübertragungen und dessen Kontrolle. Auch war es sehr interessant sich mit dem Code Composer Studio auseinander zusetzen und die vielen Möglichkeiten die dieses Tool bietet zu erforschen.

Unser letzter Dank gilt unserem Betreuer und Dozenten Herrn Schuster, der uns kompetent beraten und unterstützt hat.

9 Verzeichnisse

9.1 Abbildungen

Abbildung 1 Biometrics Market Revenues Quelle [1]	8
Abbildung 2 Basiselemente eines Fingerprintsystems Quelle [1]	9
Abbildung 3 Funktionsprinzip des kapazitiven Sensors.....	10
Abbildung 4 Fingerabdruck mit 32 Minuten	12
Abbildung 5 Equal Error Rate	14
Abbildung 6 Graustufen-Fingerabdruck.....	16
Abbildung 7 Auflageposition der 3 Templates	18
Abbildung 8 Fingerabdruck vor Filter Fingerabdruck nach Filter	19
Abbildung 9 Zwei gleiche Fingerabdrücke	21
Abbildung 10 Zwei unterschiedliche Fingerabdrücke	22
Abbildung 11 6711DSK-Board mit Daughtercard.....	25
Abbildung 12 Blockscheema der Tochterkarte	26
Abbildung 13 Code Composer Studio.....	28
Abbildung 14 GEL-Menu.....	32
Abbildung 15 Blockscheema des McBSP.....	33
Abbildung 16 Blockdiagram des DSK6711	35
Abbildung 17 Blockscheema des EMIF.....	38

9.2 Tabellen

Tabelle 1 Kleiner Ausschnitt eines Fingerabdruckes in Matrix Darstellung	17
Tabelle 2 Memory Mapping	31
Tabelle 3 Speicherbelegung der McBSP-Register	34
Tabelle 4 Register der QDMA	36
Tabelle 5 Channel Option Register des QDMA	36

9.3 Formeln

Formel 1 Berechnung des Graustufenwertes	18
Formel 2 Berechnung der relativen Differenz	21