

Besprechung der Arbeitspakete für die technische Umsetzung des Projektes.

I. Technische Beschreibung (Kann noch ergänzt werden)

Es wird ein Framework für den XMC4500 Mikrocontroller entwickelt, mit dem verschiedene Anwendungen ohne große Anpassung betrieben werden können. Als Übertragungsprotokoll dient UDP, über diese, mit dem SIP-Protokoll eine Kommunikation mit einem anderem Mikrocontroller aufgebaut wird. Die Kommunikation wird über IPsec abgesichert. Wer mit wem kommunizieren darf, wird über Open IMS geregelt.

II. Arbeitspakete

Es wurden folgende Arbeitspakete definiert (Beschreibung wurde vom Autor eingefügt und kann/wird sich noch ändern bzw. soll durch das Team konkretisiert werden):

1. AP „Open IMS“

Open IMS (IP Multimedia Subsystem) dient als zentrale um die Kommunikation z.B. zwischen einem Sensor (Temperaturfühler) und einem Aktor (Heizungsregelung) herzustellen. Dazu wurden im Raum 1.009a 4 Server bereitgestellt (Zugang: Username: student, PW: student, Root: Barker01). Auf diesen müssen noch die einzelnen Softwarekomponenten installiert, konfiguriert und evtl. programmiert werden. Open IMS benötigt einen DNS-Server, dieser muss noch auf dem Open IMS Proxy-Server installiert und konfiguriert werden (Roland).

2. AP „SIP“

Das SIP-Protokoll wird für den Kommunikationsaufbau und -steuerung benötigt und basiert auf dem UDP-Protokoll. Dabei sollen alle nicht relevanten UDP-Pakete abgelehnt und nur die relevanten Informationen im SIP-Protokoll berücksichtigt werden. Um die CPU des Mikrocontrollers zu entlasten, soll dies wenn möglich im Ethernet-Controller gefiltert werden. Laut Herrn Barker ist das mit dem Atmel nicht möglich, ob es mit dem verwendeten XMC 4500 möglich ist, muss noch recherchiert werden. Das SIP-Protokoll soll zunächst während der Entwicklung unverschlüsselt, später jedoch durch IPsec abgesichert werden.

3. AP „IPsec“

IPsec soll eine gesicherte Kommunikation über das aufgebaute Netzwerk gewährleisten. Es wurde festgelegt, dass kein Key-Exchange stattfinden soll. Der im Open IMS hinterlegte Key wird lediglich in den entsprechenden Mikrocontroller kopiert und in einer persistenten Variable abgelegt.

4. AP „Portierung Software“

Es existiert bereits ein Framework um UDP-Pakete zu empfangen. Dieses wurde aber für den Atmel Mikrocontroller entwickelt und muss nun für den verwendeten XMC 4500 Mikrocontroller

angepasst werden. Dabei wurden schon Konflikte mit dem Ethernet-Treiber festgestellt und muss noch näher untersucht werden.

III. Repository und sonstiges

Als IDE wird Keil verwendet und kann als Demo bis 32 KB genutzt und kostenlos heruntergeladen werden. Lötstation und Material wie Mikrocontroller, Kabel, ... stehen im Raum 1.009a zur Verfügung.

Da wir nur eine begrenzte Anzahl an Mikrocontroller zur Verfügung haben (8 Stk.), können die Pakete auch mittels Keil simulieren werden. Dadurch wäre man nicht vom Open IMS und der Hardware abhängig.

Für das im Punkt II.4 erwähnte Framework wurde ein GIT Repository eingerichtet. Die interessierten Informatiker und BWL' er müssen mir unter folgender E-Mail Adresse einen gültigen SSH-Key (public) zusenden.

SSH-Key rsa mit Betreff „IOT-GIT“ an die Adresse wilhelm.roland@gmx.de

Anleitung unter: <https://help.github.com/articles/generating-ssh-keys> oder mit PuTTY.

Nachdem der Key freigeschaltet wurde, erhaltet ihr eine E-Mail mit der GIT Adresse auf die ihr dann Lese- und Schreibrechte mit „clone, „pull“ und „push“ erhaltet.

Es gibt nur 1 Repository mit den verschiedenen Projekten, aus diesem Grund bitte ich euch gewisse Verhaltensregeln einzuhalten, um das miteinander zu vereinfachen.

Also bitte:

- Der Master Branch sollte immer Lauffähig und einsetzbar sein.
- Bei der Entwicklung ***immer*** einen individuellen Entwicklungsbranch anlegen.
- Erst auf dem Master-Branch mergen, wenn dieser ausführlich getestet wurde.
- Committen bitte nur diejenigen Daten die auch wirklich mit „push“ hochgeladen werden sollen, also keine object oder sonstige von Keil generierte Dateien.
- Die angelegten „.gitignore“ Dateien sollten alle nicht relevanten Dateien Filter, aber der Teufel steckt ja gewöhnlich im Detail, also falls nötig anpassen.

IV. Termine

07.05.13 Team OpenIMS Zwischenstand

30.04.13 Team IPsec Zeitabschätzung

30.04.13 Team SIP Zeitabschätzung

IV.

Das Repository ist wie folgt gegliedert:

iot	// Repo-Name
_ 3rdParty	// 3rdParty Beispiel Projekte
_ at91	// Enthält die verschiedenen Atmel-Projekte
_ arm7tdmi	// Hier Projekte für ARM7 auf Atmel z.B. uip Porjekt
_ at91sam7	// Libs und BSP für Atmel Mikrocontroller
_ doku	// Doku für CPU und Mikrocontroller
_ utilities	// Ini Dateien
_ qm	// Verwendete QP-Framwork
_ xmc4500	// Enthält XMC Projekte
_ 4500relax	// Treiber für XMC
_ doku	// Manuals für CPU und Mic.
_ rtos	// FreeRTOS
_ xmc4500	// Hier Projekte für XMC4500
_ Dokumentation	// Latex Dokumentation der verschiedenen Teams

V. Teams

Jedes dieser Arbeitspakete wird durch ein Team, bestehend aus maximal 3 Mitgliedern, umgesetzt. Je nach Anzahl der Informatiker oder Interessierte BWL' er können es auch mehr sein.

Folgende Teams wurden gegründet:

1. Team „Open IMS“
 - a. Roland Wilhelm
 - b. Matthias Riegel
 - c. Franz Lorenz
2. Team „SIP“
 - a. Ralph Schaumann
 - b. Daniel Urbano
 - c. ...
3. Team „IPsec“
 - a. Quirin Grottenthaler
 - b. Matthias Vey
 - c. ...
4. Team „Portierung Software“
 - a. Dincer Beken
 - b. Barny Porcio
 - c. Reinhard Cavetchi

Roland Wilhelm