

3.6.2019

Week9 : Embedded Programming

Prep

リージョナルレビューはBluejeansで当分は継続。
Networking週にSDプロトコルのサンプルを追加。Neil
Embedded週にMicro pythonなどモダン言語、ROSを追加

オシロがラボにないのならこれ買おう

sainsmart DSO213

<https://www.sainsmart.com/products/sainsmart-dso213-4-channels-handheld-mini-digital-oscilloscope>

Molding週

- ・Mods 2.5D機能を追加
- ・Material : Hydro-stone Dry-ston Ulu & Metal (optional)

Output週にOLED追加

Assessment documentが更新されていない問題 >> TBC

>>メモ

4/10 application and implementation (4/17 break) ミッドタームなので、それぞれのファイナルプロジェクトのスコープを確認する良い機会かもしれない。

【前半】先週のレビュー

ricoさんの質問にBasからの回答

<https://www.eevblog.com/forum/testgear/digital-oscilloscope-comparison-chart/>

スケジュール変更について

Wed May 1 -> Tue April 30 (May Day)

Wed June 5 -> Thu June 6 (Eid Al-Fitr)

今週のRecitation デザイン

・Jesus Lopez

Fablab ESNEデザイン学校 Fablab Leonから受講

<http://fab.academany.org/2019/labs/leon/students/jesus-lopez/about/>

タネをまく回転草(tumbleweed)を紡ぐ糸紡ぎ機(機能を持ったモノをつくる機械をつくる)

Neil : Final Projectのトラッキングページを作ろう

Neil : MDFは重いわりに脆く、バインダーに有毒物質を使っている可能性があるよ(スムーズな表面の仕上がりが必要な時は有用)

•Akhil (Kochi)

Mechanical engineer

<http://fab.academany.org/2019/labs/kochi/students/akhilg-babu/>

世界のファブラボを旅する

<https://www.makertour.fr>

Neil : Supply side managementをしよう。タスクではなく、決められた時間から逆算すること。

Final project

トイレ。人が入ってきたら点灯して音楽を流す。カーボンフィルターで匂いを防ぐ。

Neil : 光、音楽、カーボンフィルターだけですでに三つのプロジェクト。一番重要なものを選んで最初のスパイラルを閉じることを考えよう。

•Jobin (Oulu)

Micro electronics - material sensing

<http://fab.academany.org/2019/labs/oulu/students/jobin-varghese/>

Neil : FabAcademyには色々な専門知識を持った人が参加している。Global Issue trackerをぜひ生徒同士のコミュニケーションに使って欲しい。

Fusion360CAMでRensiのツールパスを生成し切削している

FusionのCAMでやってみたいなあ。。。

ですねー

同じく by つちや

水平垂直が出ているかのテスト

より高い精度が求められる場合は、Global error(一回四角切ってどこか別のところに行ったあとにもう一度戻ってきて四角切る)とLocal error(四角切る)を両方確認する。

Final project

モバイルデバイス用のための自転車エナジーハーベスティング
volume display

•Pascal - Surinam (LA KAZ LAB)

IT security

<http://fab.academany.org/2019/labs/lakazlab/students/pascal-vanlierop/>

1枚刃はHigh density polyethalineは良いが木材には適さない
他はエコーが酷くて聞き取れなかったです

Final project ギター

<http://plugnmake.com/news/blog/2015/05/21/fablab-amsterdam-video-interview/electric-bass-guitar/>

http://archive.fabacademy.org/2016/fablabreykjavik/students/331/final_project.html

Neil : FabAcademyトップページの検索エンジンを使って、過去のプロジェクトを参照してみることに。

レビューで紹介されているの、実際に試してみたいことがたくさんあります・・・

• John - Surinam (LA KAZ LAB)

School teacher

http://archive.fabacademy.org/2016/fablabreykjavik/students/331/final_project.html

Final project

自動的に犬を放せるリーシュ

Neil : すでにBOM(材料リスト)を載せていたり、Final projectの要件をすでに満たし始めているのは良い

• Alejandro

Industrial designer (automobile design)

<http://fab.academany.org/2019/labs/leon/students/alejandro-ausejo/>

Neil : ガントチャートを作って管理しているのは素晴らしい。Supply side management! -> Excelのガントチャートがなんかしびれます

Final project

自転車、ライト、センサー、発電？

• Jofin (Kochi)

<http://fab.academany.org/2019/labs/kochi/students/jofin-thomas/>

Spectrometer : ftir / ramar / uv-vis

Neil : 画像を他から借用した場合はクレジットを明記することが重要

Final project

垂直農場

ただつくるだけでも規模の大きいプロジェクトになるので、コラボレーションをお勧めする。

参考

<http://aquapioneers.io/>

・Francis (Echo fab)

<http://fab.academany.org/2019/labs/echofab/students/francis-lance/>

Final project

猫のための自動餌水やり機

Neil : ペリスタルティックポンプperistaltic pumpは簡単に作れる

・Jonathan - UTEC

<http://fab.academany.org/2019/labs/utec/students/jonathan-abasolo/index.html>

電動スケートボード(ロングボード)

ブラシレスモーターのドライバーを探している。

Neil : Output週 参照

[A4941](#), [A4963](#)

・Markus (Kamplimfort)

カーフベンディングを使ったツール

<http://fab.academany.org/2019/labs/kamplintfort/students/markus-fleischer/week8.html>

ジョイント部分が見えないデザインで美しい

【後半】Classまとめ

http://academy.cba.mit.edu/classes/embedded_programming/index.html

今週はマイクロコントローラーについて学ぶ週

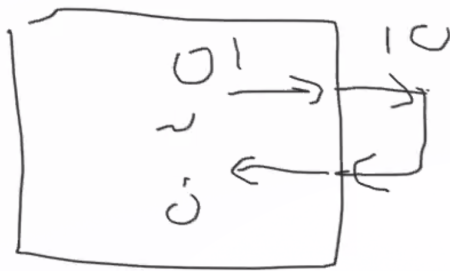
今週のゴール

プログラムをマイコンにアップロードする基本的なワークフローになれること

ハーバードアーキテクチャ

プログラムとデータは別のメモリに入っている

一番最初のコンピューターのバグは、実際の虫がコンピューターの中に入ってしまったことによる。これがバグという用語の起源。

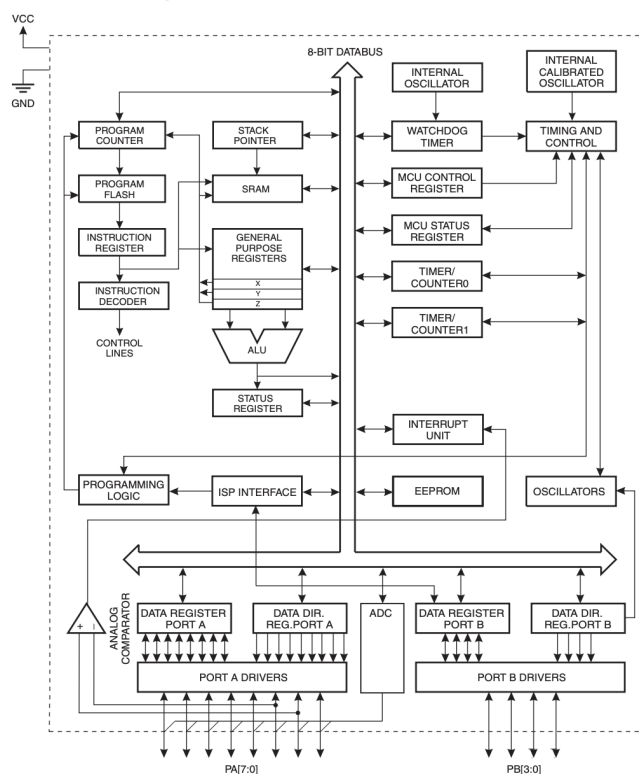


↑マイコンの速さを測った方法。

ノイマン型コンピュータ
プログラムとデータが一緒の場所にある

全てを記憶する必要はないが、何が書いてあるかを把握して欲しい
[data sheet](#)

Figure 2-1. Block Diagram



ブロックダイアグラムに書かれたコンポーネント一つ一つを説明していく。

- Flashメモリ >> プログラムが書き込まれるメモリ
- Fuseメモリ
- I/Oポート

Figure 10-2. General Digital I/O⁽¹⁾

The diagram illustrates the internal logic of a General Digital I/O pin. It shows the connection between the pin, internal pull-up/pull-down transistors, and various control and data registers. The pin is connected to a pull-up transistor (controlled by PUD) and a pull-down transistor (controlled by SLEEP). The pin signal is buffered and connected to the D0n and D0Pn registers. The D0n register is controlled by RESET and WDx, and its output is RDx. The D0Pn register is controlled by RESET and WRx, and its output is RPx. The D0Pn register is also connected to the WPx register. The D0n and D0Pn registers are connected to the Synchronizer, which is controlled by SLEEP and provides the I/O clock (clk_{IO}). The Synchronizer is connected to the D0n and D0Pn registers. The D0n and D0Pn registers are connected to the WPx register. The D0n and D0Pn registers are connected to the WPx register. The D0n and D0Pn registers are connected to the WPx register.

Legend:

- PUD: clk_{IO}
- PULLUP DISABLE
- SLEEP: SLEEP CONTROL
- IO CLOCK
- WDx: WRITE DDRx
- RDx: READ DDRx
- WRx: WRITE PORTx REGISTER
- RPx: READ PORTx REGISTER
- RPx: READ PORTx PIN
- WPx: WRITE PIN REGISTER

・timer / PWM : 実際に電圧を帰るのではなく、ポートを高速ON/OFFすることで同じ機能を実現する(熱効率など)

一番プリミティブな8bitマイコン、一度に扱える情報量は8bitだが20MHzで動いているので実際にはもっと多くの情報を扱える。(例えばオーディオ情報16kHzなので十分扱える?)

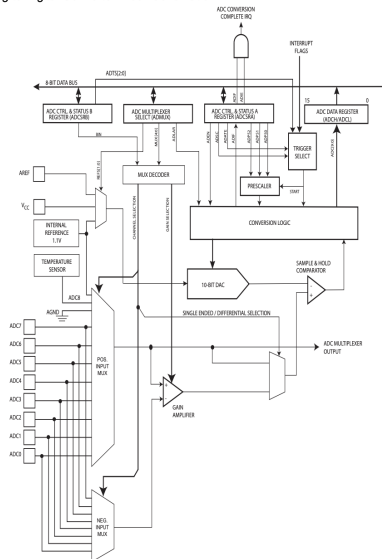
AVRに焦点を当てるのは、toolchainがたくさんあるから
アーキテクチャに合わせてコンパイラをつくるのではなく、AVRは既存のコンパイラに合わせて設計されたアーキテクチャ。それがAVRのtoolchainが充実している理由。

Attiny10 (米粒大)から始まり、
数字が大きくなればなるほど、ピン数などリソースが増える。

<http://www.ladyada.net/learn/avr/>

- ・A/Dコンバーター

Figure 16-1. Analog to Digital Converter Block Schematic



データシートを斜め読みする。

マイコンのプログラミングにおいては、ハードウェアを理解しない限りソフトウェアを理解することはできない。

→People who are really serious about software should make their own hardware. アラン・ケイのこの言葉を思い出しました
なるほど

・クロック

RCクロック回路: マイコンの中に入っている。しかし、精度は10%、調整すると1%。
セラミック(レゾネーター)や水晶を外部につけることでより正確なクロックを得られる。

・ISP

pads >> ヘッダーを半田付けせずに、プログラムするときだけ押し付ける。
clips >> マイコンのピンに直接クリップする道具
など

・ブートローダー

・プログラマー

Jtags >> ARMS

PDI >> Atmel

hex fileが最終的にマイコンに書き込むコード

instruction set

アセンブラ

C

マイコンに書き込むプログラムはだいたいC言語で書かれる。

Neil : 学び始めるときは一からコードを書くのではなく、サンプルコードを編集するところから始めると良い。

GCC コンパイラ

AVR - libc 演算などのソフトウェアライブラリ

Bus : WinAVRはもうサポートされてないので避けよう

ホストコンピューターとの通信

RS232 昔からある成熟したプロトコル

ホスト側では送受信をするためのターミナルが必要

[Screen](#) [Kermit](#) [Minicom](#) [Arduino](#)

[pySerial](#) [miniterm.py](#) [rx.py](#) [term.py](#)

[SerialPort](#) [terminal](#)

USB

FTDI(社名)

USBからシリアル信号(TTL ロジックレベル)に変換してくれる変換器

modsでもシリアル確認できる。。。

echo hello-world

・[hello.ftdi.44.echo.c](#)

ホスト側で打った文字をそのまま返す

・[hello.ftdi.44.echo.interrupt.c](#)

割り込みを待つ

・[hello.ftdi.44.echo.asm](#)

全く同じ内容をアセンブラで書いたもの(メリットは書き方によっては効率化できるのと、コードから処理にかかる時間が読み取れる)

※外部ライブラリは使っていないみたい。

使わない方がいい>処理早いみたいなこと言ってましたね、、、、と、、言われても、、(泣)

XMEGAファミリー

クロックがより早いライン。49Mhz.さらに高速高度なことができる。

[Adafruit](#) [Olimex](#) [Pololu](#)

安価な既製品マイコンボードのメーカー

FabAcademyでは、理解を深めるためにマイコンボードを自分で設計して制作するが、アカデミー後は使う選択肢に入れよう

Arduinoの5つの要素:

- ヘッダ
- ブートローダー
- ボード
- C言語ライブラリ
- IDE

これらは別々に組み合わせて開発することができる。

ブートローダーを一度書き込んだら、ISPヘッダ経由でプログラムする必要はない。
IDE>シリアル通信で書き込みができるようになる。

Arduinoでプログラムする=Cでプログラムする、という判定になりますか？(課題)
↑Cでやったことにはならないんですが、マイコンプログラムの方法の一つとしてカウントされます。今週の課題はArduinoIDE/Arduino言語でAVR(Attiny)のプログラムがで最低限できればクリアです。
ArduinoIDEで純Cを書き込んでもらってももちろん構いません！
→わかりました！

あ、すみません、1個だけだとダメかも、、、GCCツールチェーンでの書き込みもやる。(ニールのサンプルちょい替えとか)

例)

[ATtiny](#)

ArduinoIDEで、Attinyシリーズで開発する例

ARM

- ・nRF52 無線通信機能のついたチップ

↑これ去年からNeil推しますよねえ。。ですね

- ・ラズパイ
- ・マイクロビット

チップの速度はソフトや使用言語によって異なる。例えば、CでプログラムしたラズパイはXmegaよりも遅い。

一般にARMはAVRより遅い。

↑そうなのか、、、

インタプリタ言語(ハイレベル言語)

- ・対話型でプログラムできる
- ・パフォーマンス(速度)を犠牲にする

=====

今週の課題について

- ・データシートを読む

全て記憶しなくても良いが、隅から隅まで目を通すこと(全て重要)

- ・ボードをプログラムする。

以前に作ったボードにプログラムを書き込む

ニールのサンプルプログラムを部分的に書き換えることから始める

可能な限りいろんな種類のプログラム言語、開発環境を試す。わからなければ、とりあえず ArduinoIDEを使ってみること。余裕があれば、他のものにも手を出してみる。

プログラムを書き込んで何も起きなかったときどうするか？

デバッグ

-目視でのチェック(はんだ不良とか)

-オシロスコープなどで電圧をチェックする 信号がきているかどうか

最後Global Good byeをそれぞれの言語でやったのがよかった。
お疲れ様でした。