CYBERCOM GROUP

Boosting your performance in the connected world.

Hobbyelektronik med Raspberry Pi och Erlang

Agenda

Det här är berättelsen om en resa,
så därför blir presentationen som en reseberättelse.

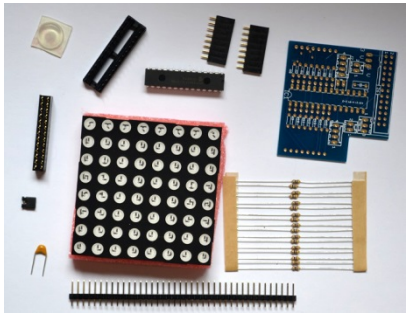
1. Var ska man börja?
2. De första stegen mot något eget
3. Gör det ordentligt!
4. Andra saker jag gjort
5. Dags att hitta på något eget...
6. Demo

Följ med på resan!

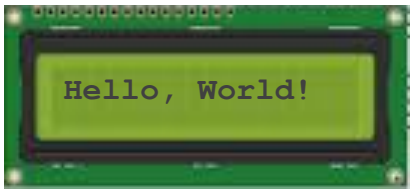
1. Var ska man börja?

- Alla har vi varit nybörjare
- Det finns gör-det-själv-projekt på:
  - Adafruit
  - ModMyPi
  - SparkFun ...

  ”IKEA-byggsats”

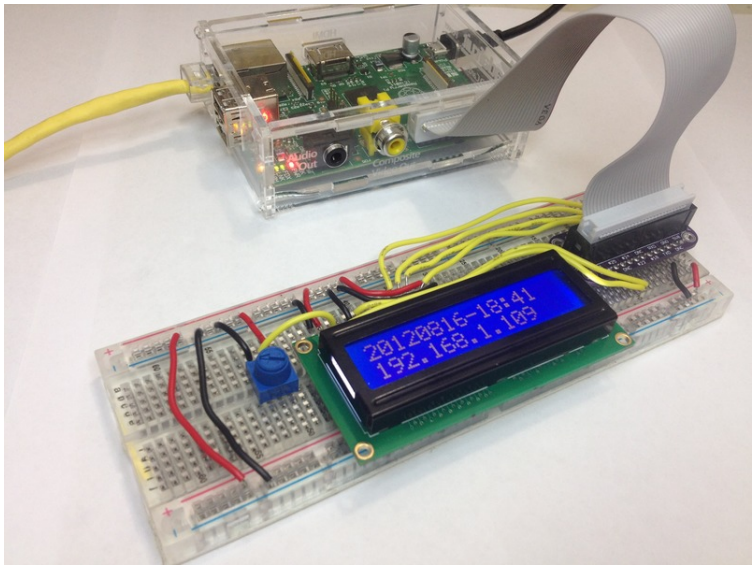


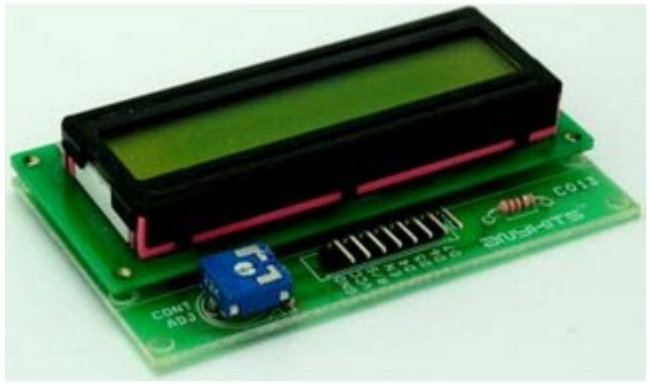
- Enkelt och ”häftigt” att bygga eget
- ”Användbart” och lätt att modifiera koden

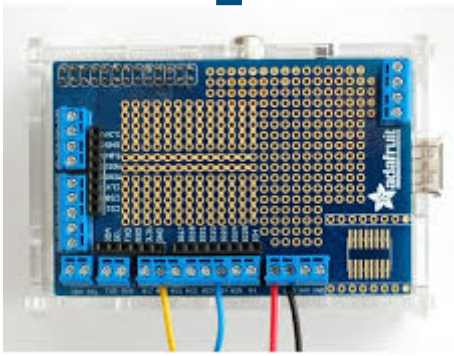


2. De första stegen mot något eget

- Byt ut någon komponent

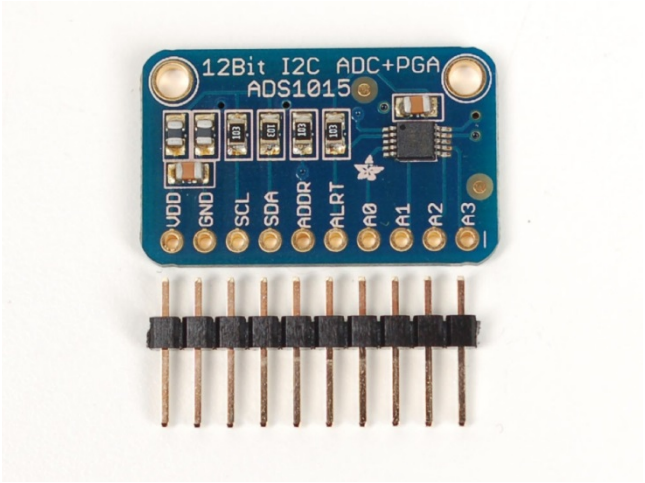


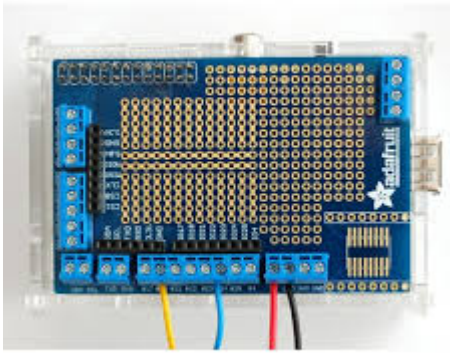




2. De första stegen mot något eget

- Kombinera

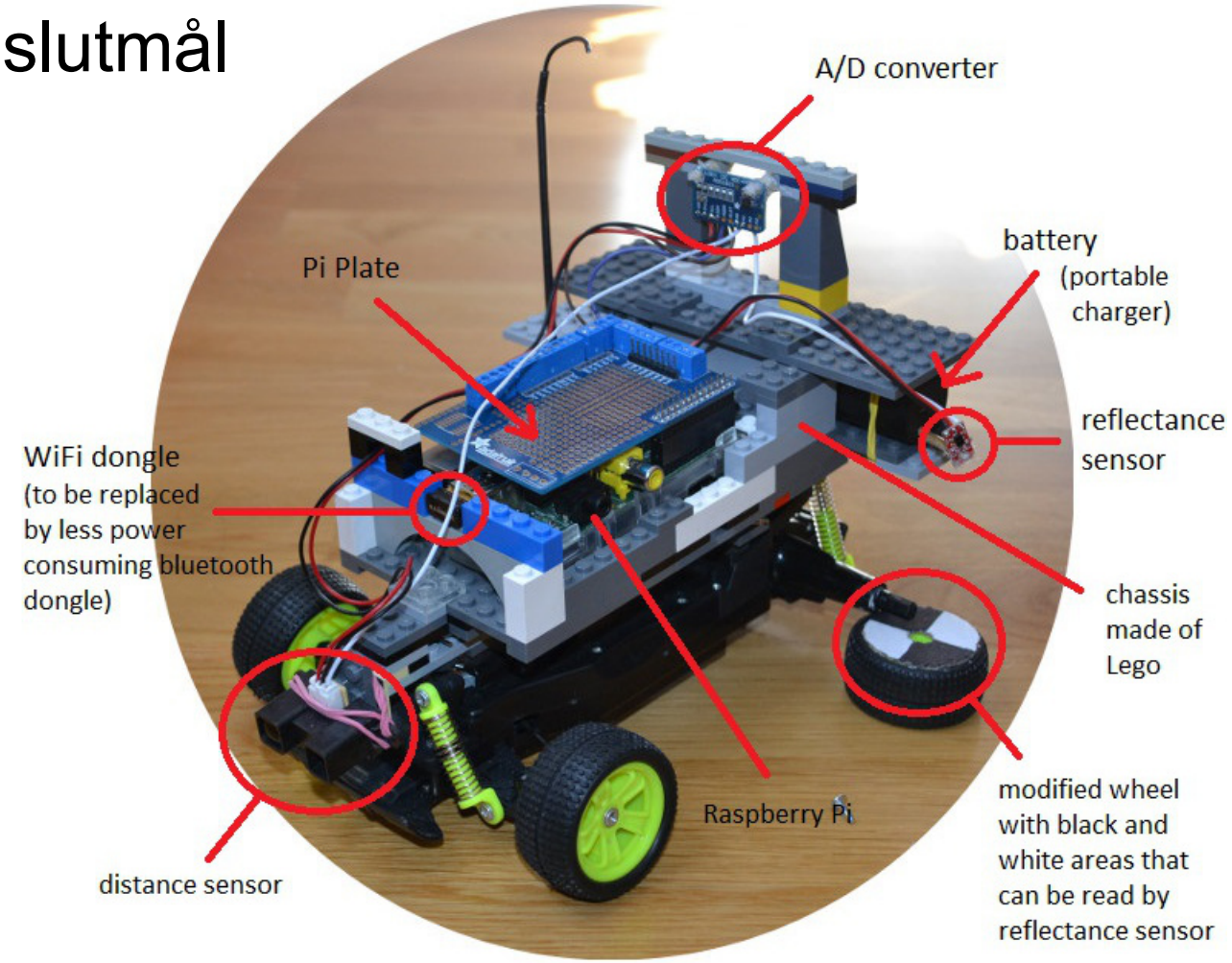




Datablad

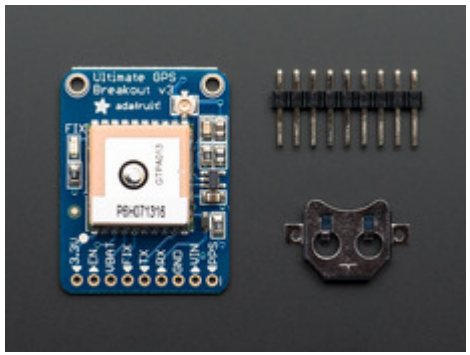
2. De första stegen mot något eget

- Ett slutmål



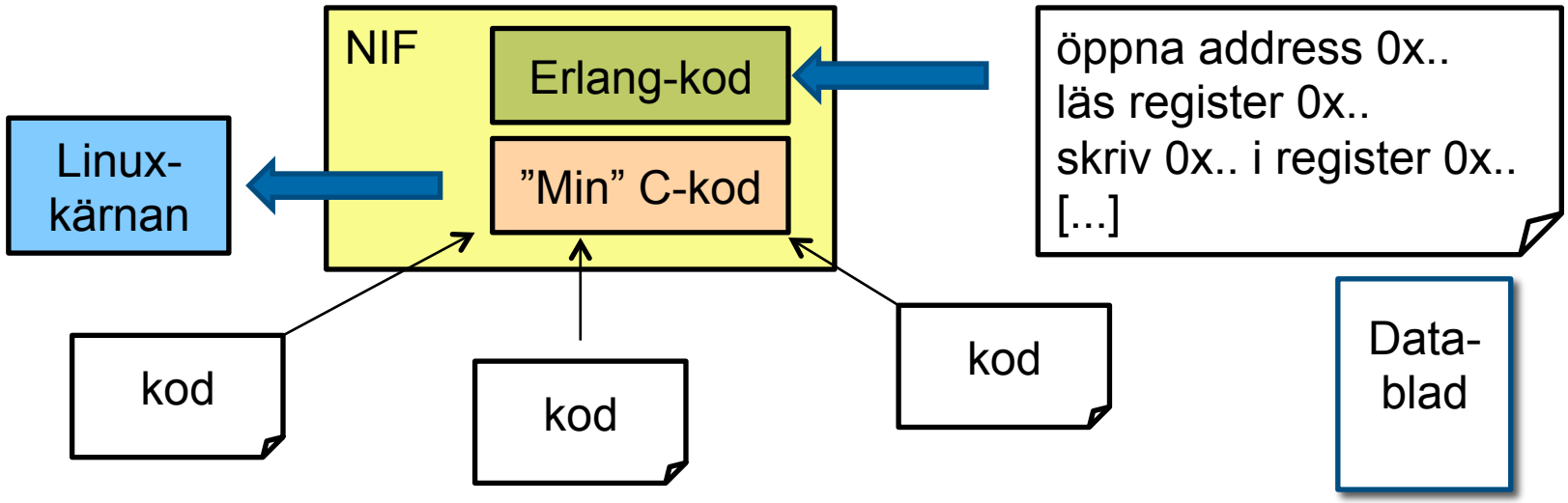
2. De första stegen mot något eget

- Så vad hände?
  - Avståndssensorn var omöjlig att kalibrera, eftersom felmarginalen var större än det uppmätta värdet – antagligen för ljuskänslig
- Tidigare tanke:
  - GPS för att mäta hastighet
    - ”funkar” över UART – *men*
    - låg uppdateringsfrekvens
    - inte tillräckligt noga, få satelliter



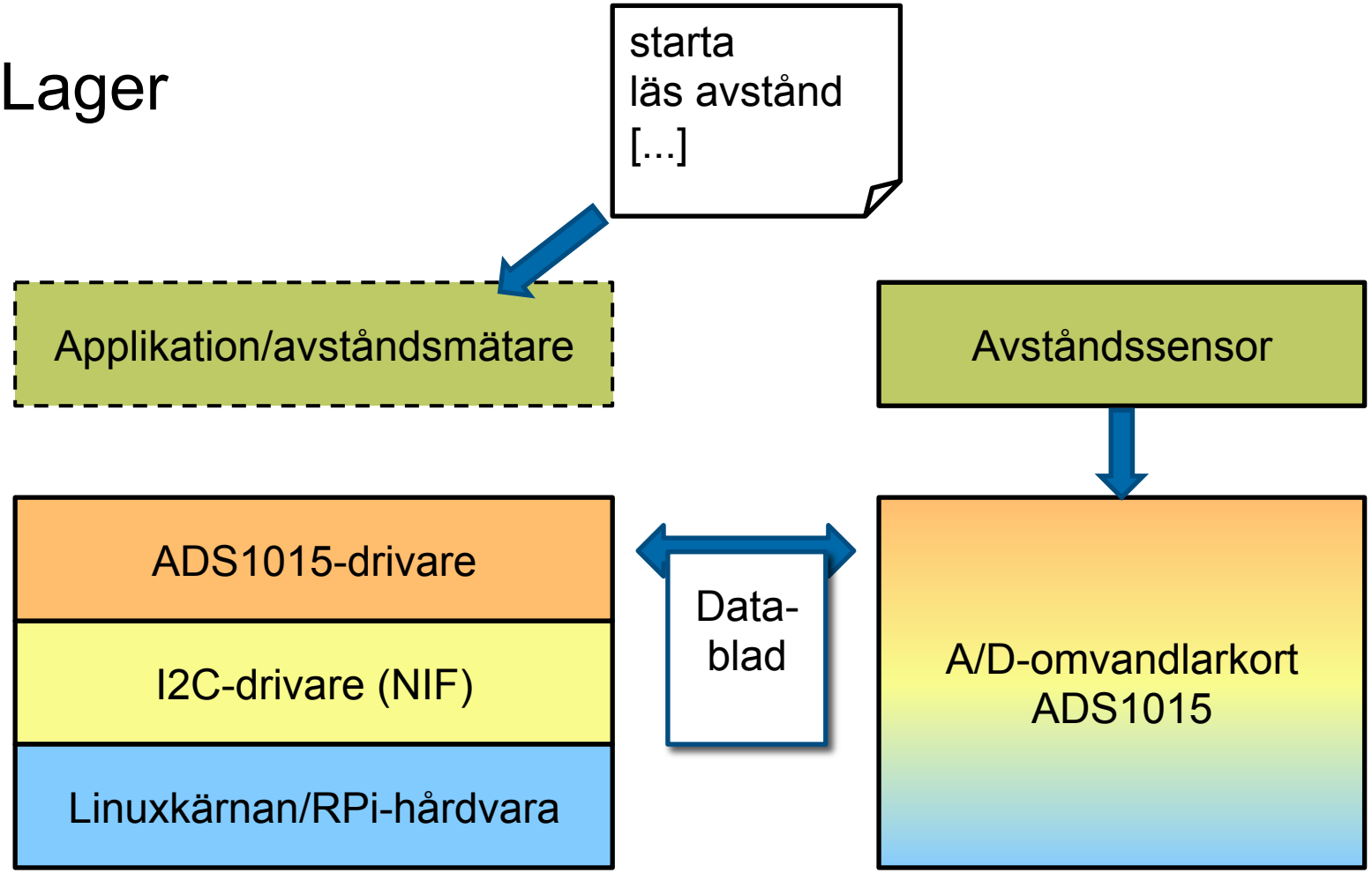
2. De första stegen mot något eget

- Dags att skriva Erlang-kod
  - Google: ”I2C Linux”
    http://elinux.org/Interfacing_with_I2C_Devices
    https://www.kernel.org/doc/Documentation/i2c/dev-interface
  - SMBus? Kanon, det tar vi.



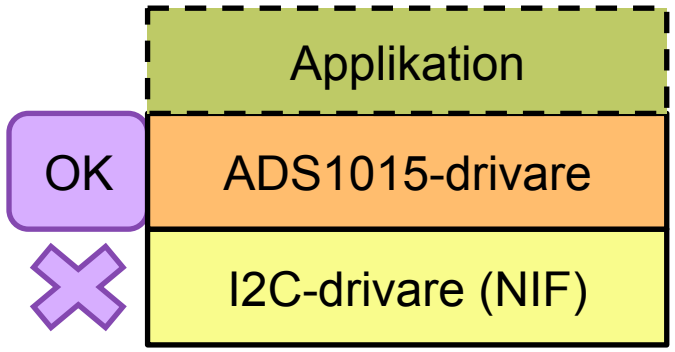
3. Gör det ordentligt!

- Lager



3. Gör det ordentligt!

- Testdrivet
  - Enhetstester ”the Erlang way”: EUnit
    - Mockgyver
    - Proper



Mockgyver – ett exempel

CYBERCOM GROUP

```
setup() ->
   [...]
   %% Mock I2C interface methods
   ?WHEN(i2c_interface:open_i2c_bus(_Address) -> {ok, ?HANDLE}),
   ?WHEN(i2c_interface:close_i2c_bus(_Address) -> ok),
   [...]
   {ok, Pid} = ads1015_driver:start_link(),
   Pid.

[...]

init_should_open_i2c_bus_test(_) ->
   ?WAS_CALLED(i2c_interface:open_i2c_bus(?I2C_ADDRESS)).

terminate_should_close_i2c_bus_test(Pid) ->
   ads1015_driver:stop(),
   wait_for_exit(Pid),
   ?WAS_CALLED(i2c_interface:close_i2c_bus(?HANDLE)).
```

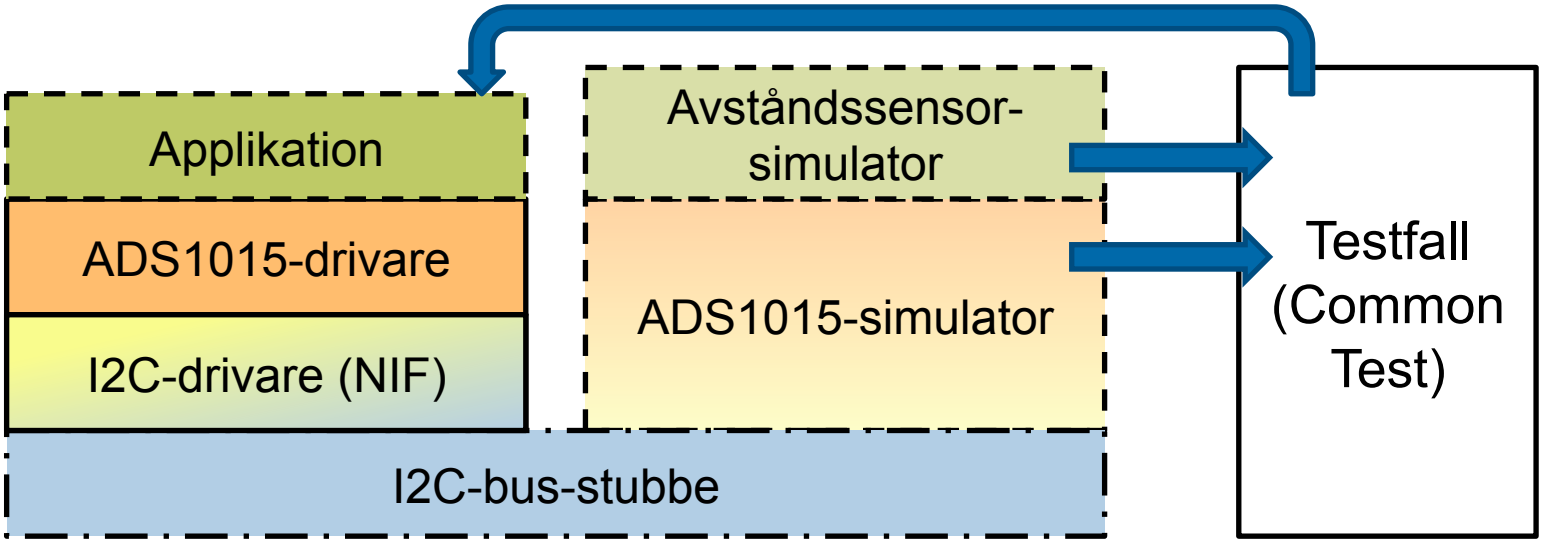
Proper – ett exempel

CYBERCOM GROUP

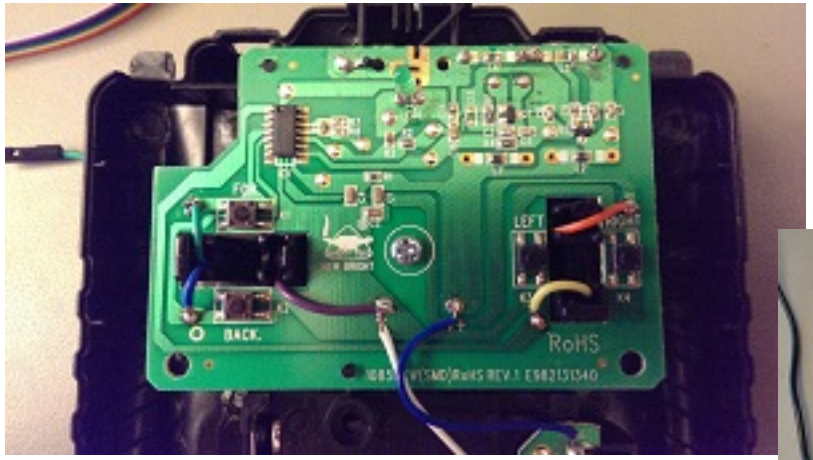
```
prop_set_status_bit_always_sets_status_bit() ->
   ?FORALL(BitPattern, word_value(),
           begin
              NewBitPattern =
                 ads1015_driver_lib:set_status_bit(BitPattern),
              is_integer(NewBitPattern)
                 and ((NewBitPattern band ?STATUS_BIT) > 0)
           end).

[...]

prop_decodes_encoded_data_rate() ->
   ?FORALL(DataRate, data_rate_value(),
           DataRate ==
              ads1015_driver_lib:decode_data_rate(
                 ads1015_driver_lib:encode_data_rate(DataRate))).

data_rate_value() ->
   oneof([128, 250, 490, 920, 1600, 2400, 3300]).

[...]
```

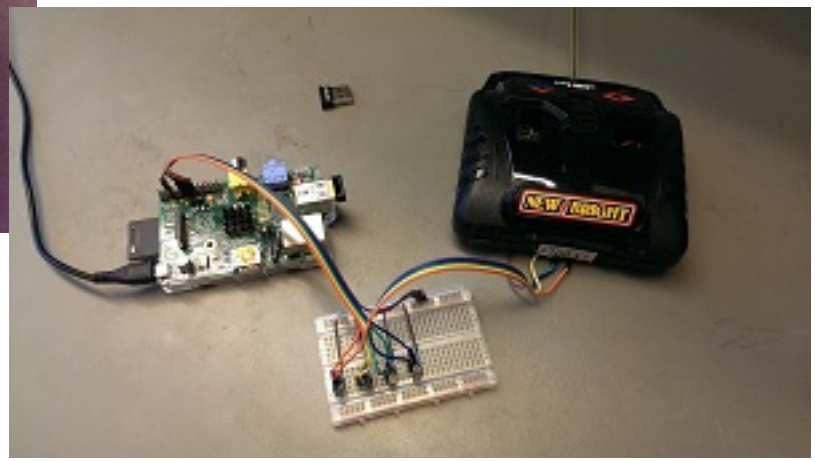
3. Gör det ordentligt!

- Önskemål: Systemtester med Common Test
  - ”The Erlang way”:
    - Simulatorer: Enkelt att skriva i funktionella språk
    - Meddelanden: Lätt att skicka i Erlang



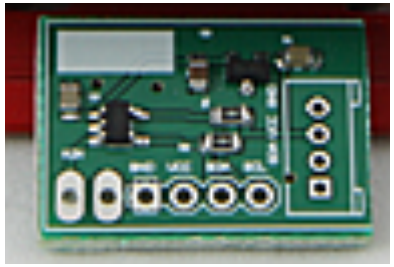
4. Andra saker jag gjort



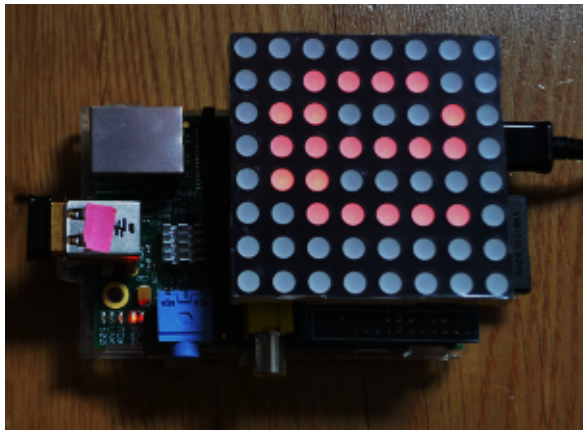


4. Andra saker jag gjort

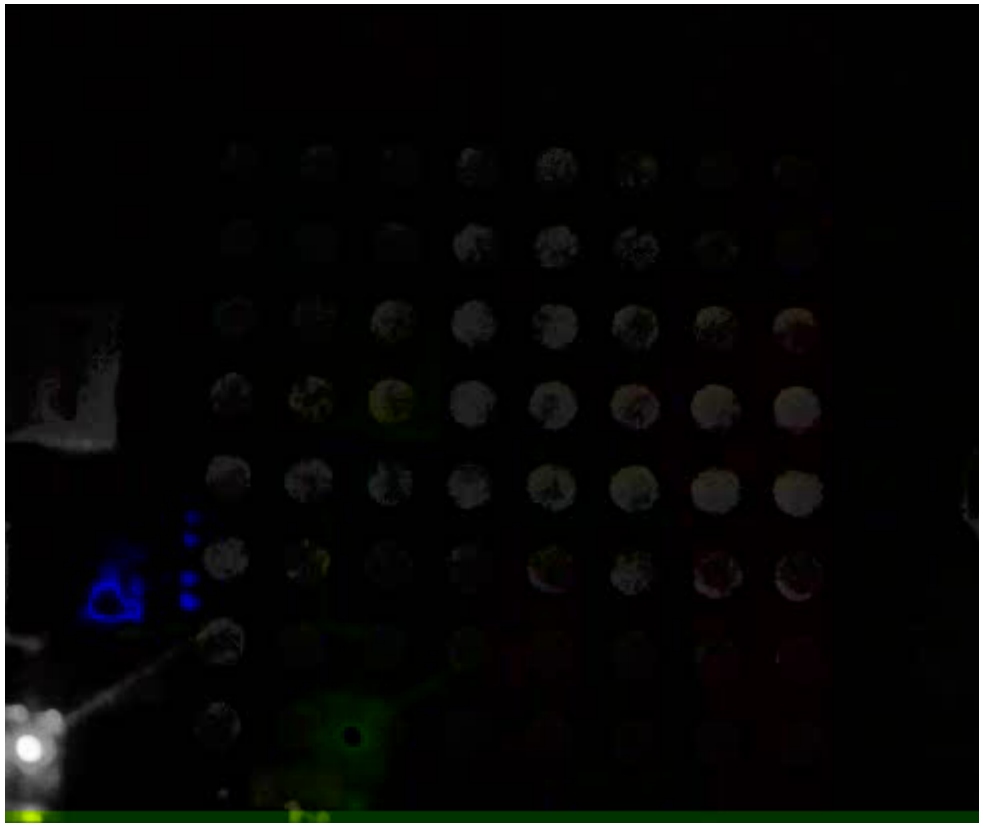
- Bytt ut A/D-omvandlaren mot en enklare variant
  - 1 kanal istället för 4
  - Ingen komparator
  - Snabbare svarstider
  - Pratar ”rå” I2C
  - Behöver ingen logik eller drivare



- Programmerat en LED-matris
  - Åtkomst via rader och kolumner
  - Kräver en ”svepning”
  - Svepning ger annan intensitet
  - Lätt att göra animeringar



4. Andra saker jag gjort



4. Andra saker jag gjort

- Blåtand
  - Google: Erlang bluetooth – ”inga” träffar
  - Skriv en NIF mot Bluez
  - Skicka Erlang-termer som binärer
  - Använd RFCOMM som protokoll (”serieport”)
  - Köp ett kort med rätt profil och räckvidd
  - Kortet kräver ”aktivering” (scan mode)
  - Socketen ”fungerar” som en TCP-socket
  - Kräver ”pakethantering” utan överliggande protokoll eftersom Erlang inte ”ser” när den tar emot flera termer
  - Målsättning:
    - Skriv ett ramverk som inets (bnets)
    - Hantera tjänster

Blåtandsserver – ett exempel

CYBERCOM GROUP

```
go() ->
   {{local, LocalMac}, {remote, _RemoteMac}} =
      get_mac_addresses(),
   {ok, Socket} = bluetooth_interface:create_rfcomm_socket(),
   ok = bluetooth_interface:bind_bt_socket(Socket, ?PORT,
                                           LocalMac),
   ok = bluetooth_interface:bt_socket_listen(Socket),
   Pid = spawn_link(?MODULE, socket_acceptor, [self(), Socket]),
   receive
      {Pid, done} -> ok
   after 60000 ->
      error(timeout)
   end,
   bluetooth_interface:close_bt_socket(Socket).

socket_acceptor(Caller, Socket) ->
   {ok, Socket2, RemoteAddress} =
      bluetooth_interface:bt_socket_accept(Socket),
   receive_loop(Socket2),
   Caller ! {self(), done},
   ok.
```

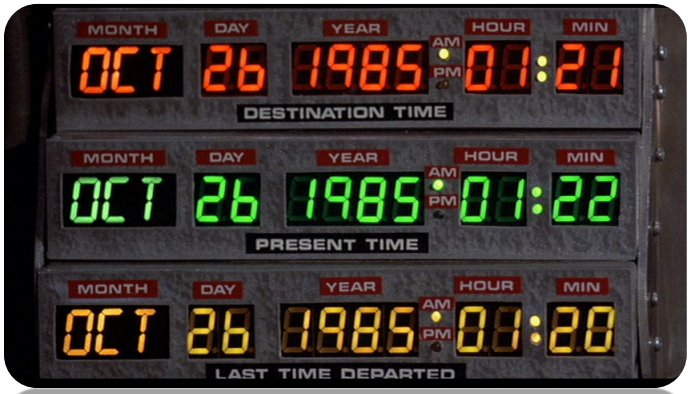
Blåtandsklient – ett exempel

CYBERCOM GROUP

```
go() ->
   [...]
   {ok, Socket} = bluetooth_interface:create_rfcomm_socket(),
   Pid = spawn(?MODULE, socket_connector, [self(), Socket,
                                           RemoteMac]),
   receive
      {Pid, done} -> ok
   end,
   bluetooth_interface:close_bt_socket(Socket).

socket_connector(Caller, Socket, RemoteMac) ->
   ok = bluetooth_interface:bt_socket_connect(Socket, ?PORT,
                                              RemoteMac),
   Data = erlang:term_to_binary({self(), greetings}),
   ok = bluetooth_interface:bt_socket_send(Socket, Data),
   [...]
   Data2 = term_to_binary("Bye!"),
   ok = bluetooth_interface:bt_socket_send(Socket, Data2),
   timer:sleep(10000),
   Caller ! {self(), done},
   ok.
```

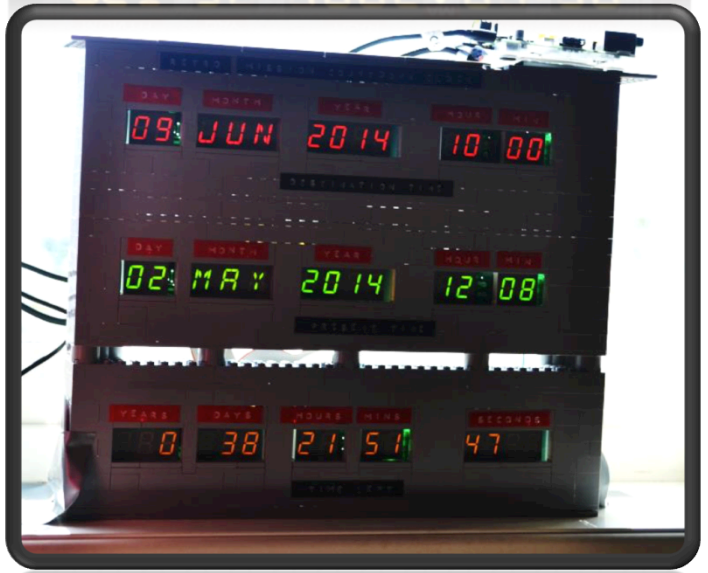
5. Dags att hitta på något eget...

- Tillbaka till framtiden



Hittad på Adafruit:

?

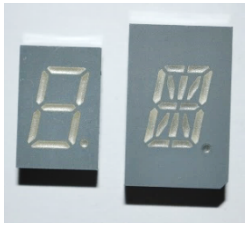


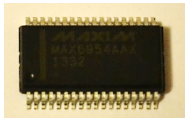
DIY!

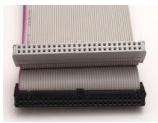
Nja... Jag vill ha *bokstäver*
– och funktionalitet!

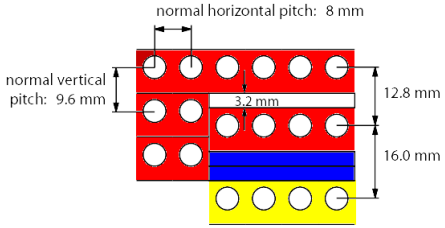
5. Dags att hitta på något eget...

- ”Hur svårt kan det va’?”
  - Det visade sig: Faktiskt inte så svårt alls...
  - 7-segments och 16-segments-displayer
    - Samma teckenstorlek
    - I tre olika färger
    - Farnell har det jag behöver
  - En kraftfull drivkrets - MAX695X
  - Mikro-USB kontaktdon, stiftlistor, flatkabel”prylar”
  - Motstånd och kondensatorer
  - Lego

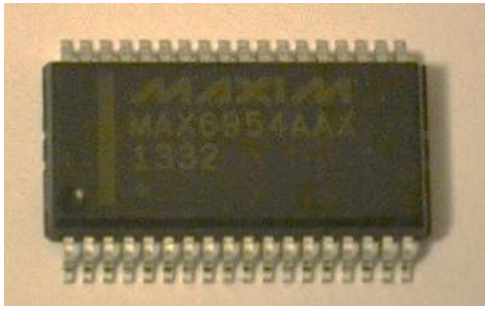








5. Dags att hitta på något eget...

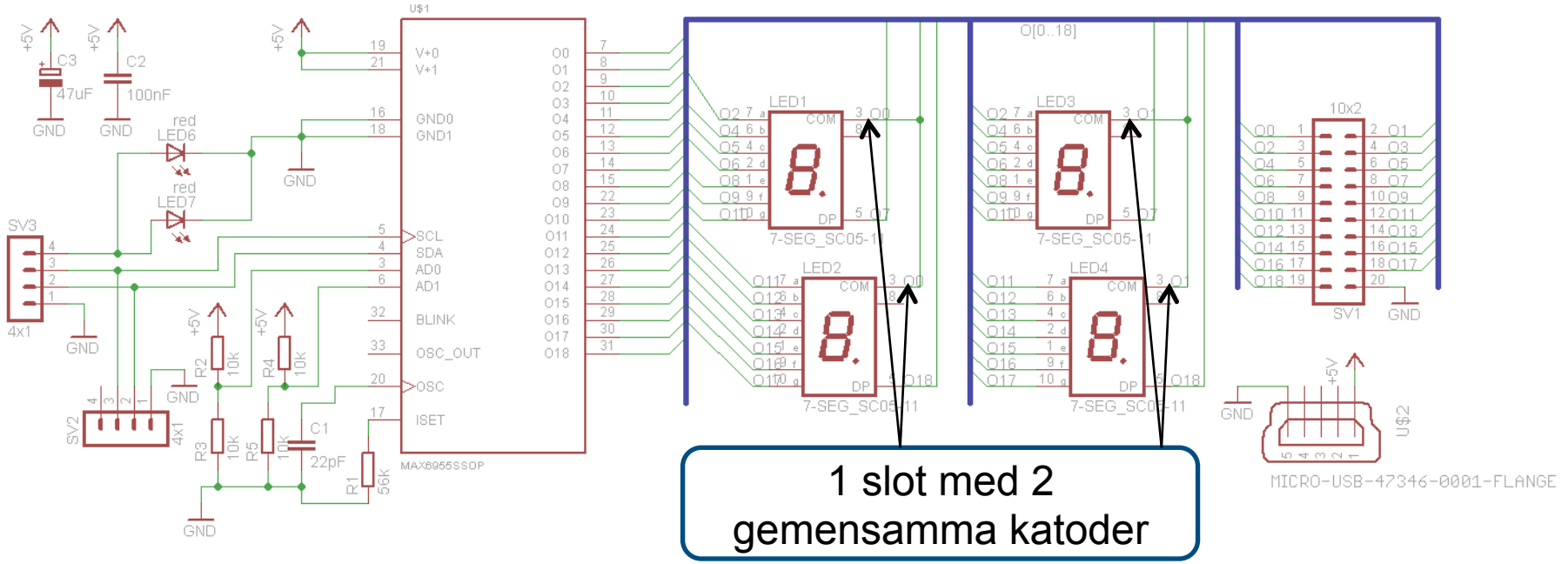


- Drivkretsen: Displayavkodare
  - MAX6955AAX
  - I2C-varianten, 2 pinnar plus jord
  - SSOP (0,8 mm mellan benen)
  - Driver 7-, 14- eller 16-segmentsdisplayer
  - 128 segment per krets, dvs en krets per rad
  - 19 output-pinnar
  - ASCII-teckentabell för 14- och 16-segmentsdisplayer
  - Även avkodning för 7-segmentsdisplayer
  - Man behöver bara ett register per tecken
  - Tecknen uppdelade i ”slots” (4x7, 2x16 eller 2x7+1x16)

5. Dags att hitta på något eget...

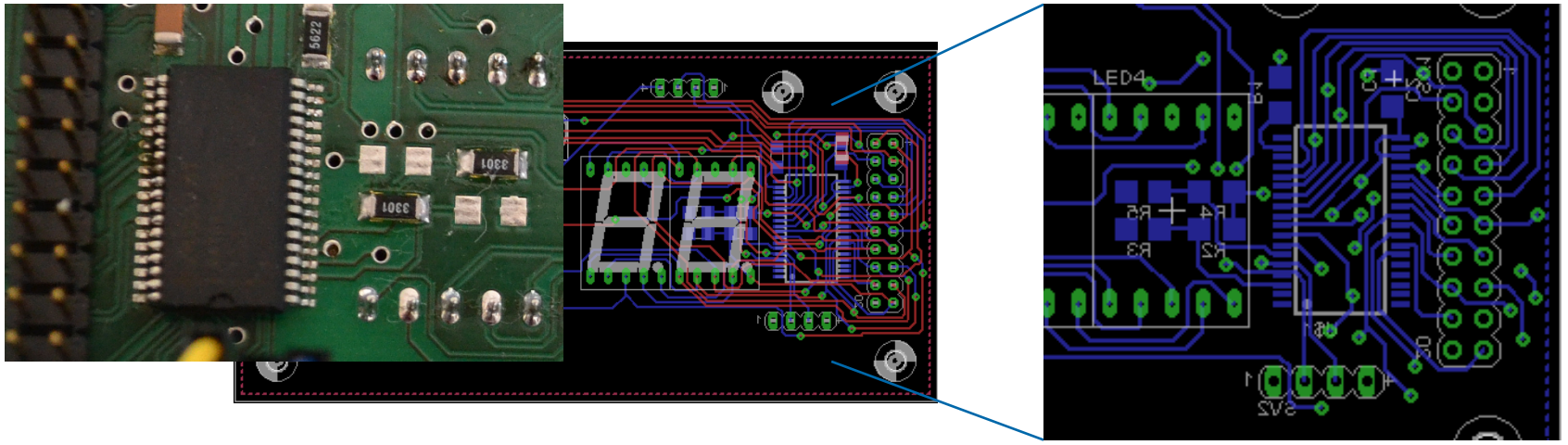
- Kretskort förstås
  - Lära sig CAD:a i Eagle
  - Läs databladet noga
  - 3 kretskort per rad, MAX-kretsen på ett av dem
    - Även strömförsörjning (mikro-USB)

Data-blad



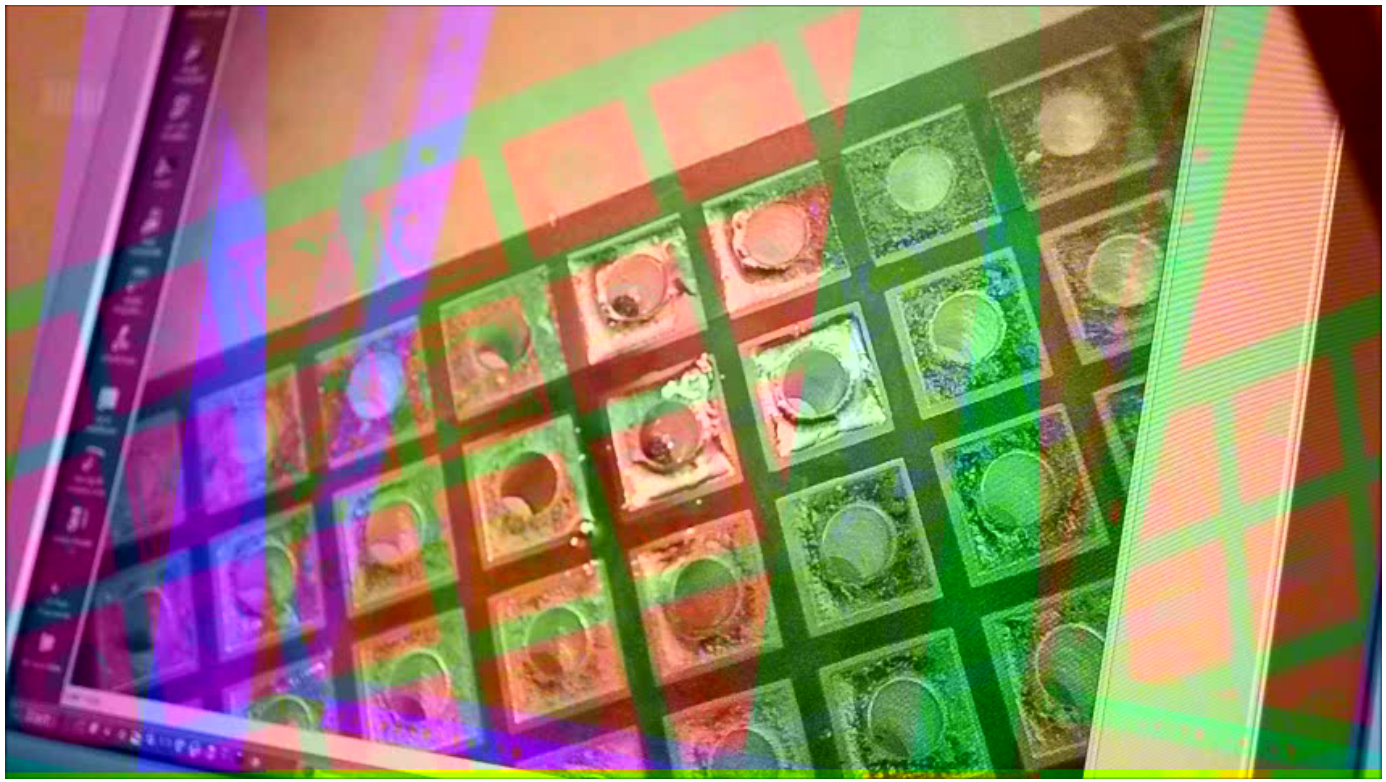
5. Dags att hitta på något eget...

- Outputpinnar blir anslutna horisontellt med flatkablar
- I2C-bussen (SDA och SCL) går vertikalt
- Monteringshål anpassade till Lego
- Inte så svårt i sig, men jobbigt att route:a
  - Väldigt många anslutningspunkter
  - Vinkeln ändras när man flyttar ledare i Eagle
  - Det blir väldigt trångt runt MAX-kretsen, många vior



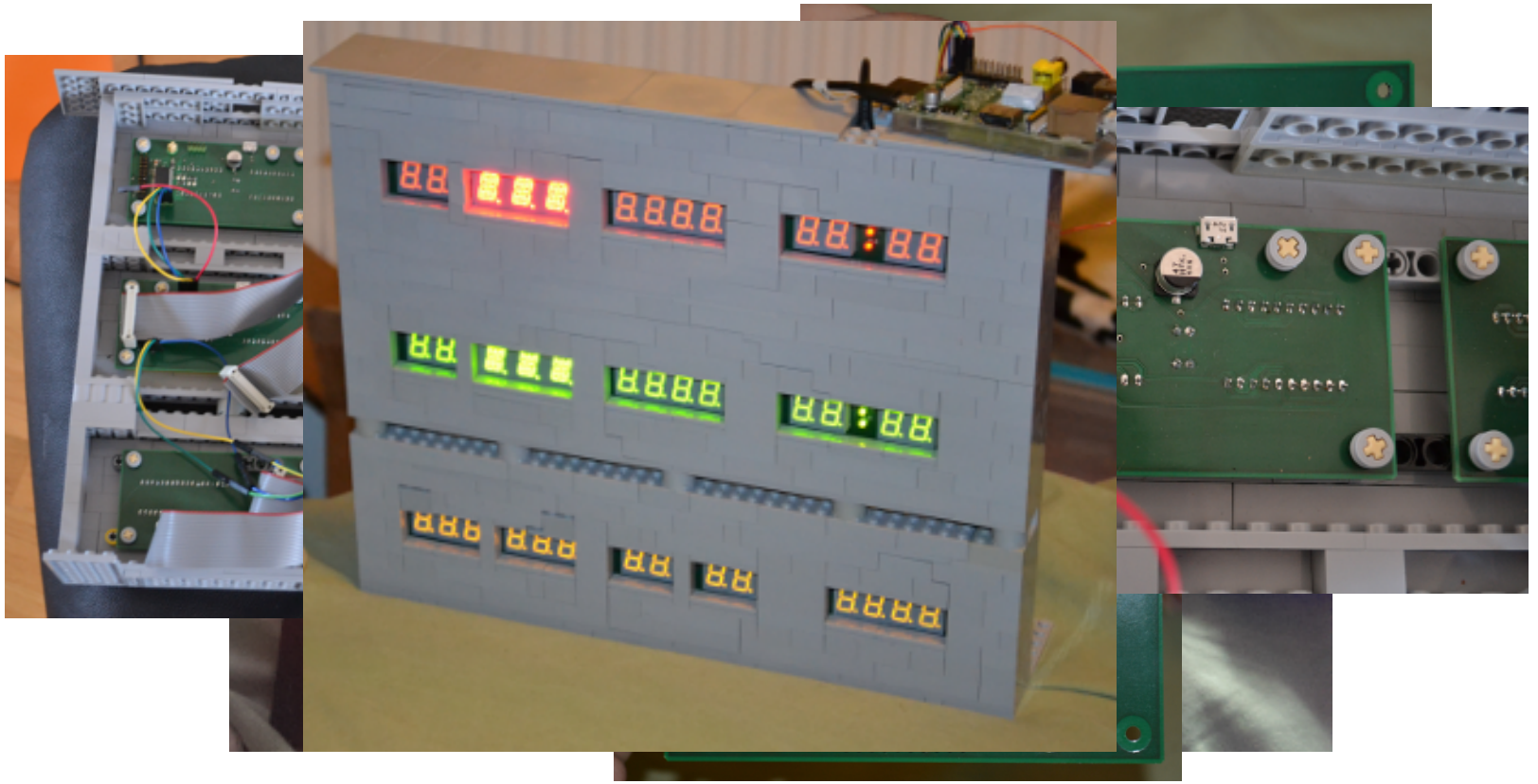
5. Dags att hitta på något eget...

- Varmluftslödning med mikroskop
  - Okej, så jag fick lite hjälp



5. Dags att hitta på något eget...

- Från prototyp till färdig produkt

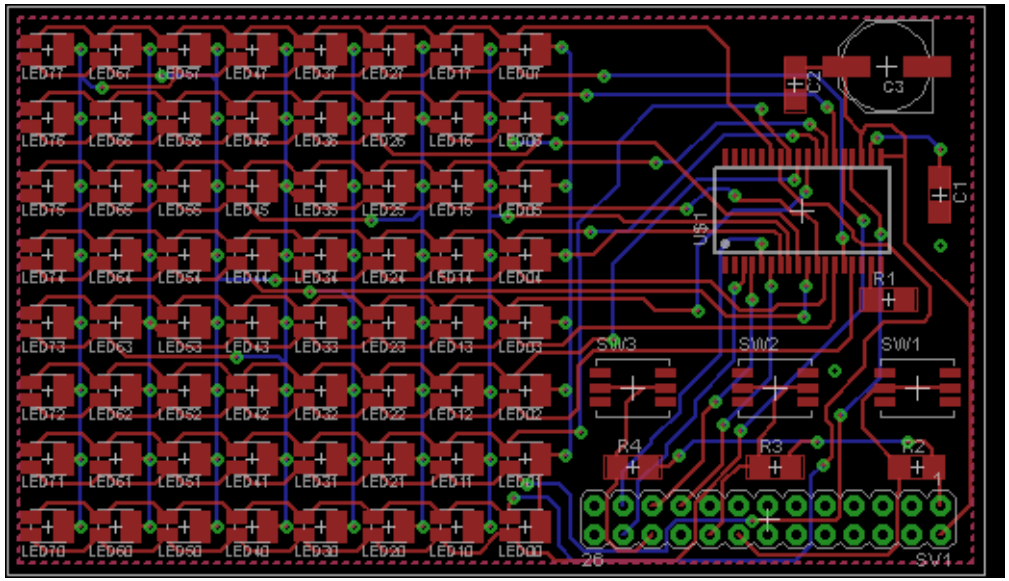


5. Dags att hitta på något eget...

- Mjukvaran
  - ”Ihopslängd”
  - Tre stycken oberoende *gen_servrar* och ett lib:
    - `destination_time`
    - `current_time`
    - `countdown_clock`
  - Delar av lib:et borde bli drivare
  - `destination_time` notifierar `countdown_clock`
  - Den ska kunna fortsätta räkna åt andra hållet
  - Man ska kunna mata in sluttid
    - ”matrix keypad”
    - kräver I/O-expander

5. Dags att hitta på något eget...

- På gång:
  - LED-array (tvåfärgad: röd och grön)
  - Drivs av SPI-versionen av MAX-kretsen
  - Är en ”sköld”
  - Har ”programmerbara” tryckknappar



6. Demo

CYBERCOM GROUP

cybercom.com