

GPIO

Tim Ackermans & Dominic Voets

Testen GPIO met Peek And Poke

Om te testen of we de joystick kunnen uitlezen moeten we eerst de GPIO aanzetten. Dit kan via het P2_MUX_SET register:

33.4.7 Port 2 multiplexer Set register (P2_MUX_SET - 0x4002 8028)

The **P2_MUX_SET** register is a write-only register that allows configuring selected pins for one of two functions. One bit in this register, EMC_D_SEL enables all EMC data lines, the other bits in this register select either an alternate GPIO-type function or an alternate peripheral function. The EMC data pins can be used as general purpose GPIO when 16 bit SDRAM or DDRAM is used.

Note: this register also controls three Port 3 pins; one GPO and four GPIO pins.

Writing a one to P2_MUX_SET bits 0, 1, 2, 4 or 5 configures the GPO or GPIO pins as the alternate function.

Writing a one to bit 3 in the P2_MUX_SET register results in all of the corresponding EMC_D[31:19] pins being configured as GPIO pins P2[12:0].

Writing a zero to a bit in P2_MUX_SET has no effect.

Door bit 3 op 1 (0x8) te zetten worden de pins geconfigureerd als GPIO:

```
# insmod /home/upload/peek_and_poke_kernel.ko
/sys/kernel/E56/hw created
# echo "w 0x40028028 0x8" > /sys/kernel/E56/hw sysfile_write (/sys/kernel/E56/hw
) called
buffer: w 0x40028028 0x8

write 8 to f4028028
```

Om te bepalen of een pin input of output is gebruiken we het P2_DIR_SET register:

32.5.18 Port2 and Port3 Direction Set Register (P2_DIR_SET - 0x4002 8010)

The P2_DIR_SET register is a write-only register that configures the data direction of GPIO[5:0] pins and EMC_D[31:19] (P2.0 through P2.12) when they are not used as EMC data lines. P3_DIR_SET is used in conjunction with P3_DIR_CLR.

Door hier 0x0 naar te schrijven worden alle pinnen op input gezet:

[1] Bit level definitions: 1 = set pin as output, 0 = set pin as input.

```
# echo "w 0x40028010 0x0" > /sys/kernel/E56/hw sysfile_write (/sys/kernel/E56/hw
) called
buffer: w 0x40028010 0x0

write 0 to f4028010
```

Na deze acties kunnen we de joystick uitlezen met het P2_INP_STATE. Zoals omschreven in de opdracht gaat het hier om EMC_D[23:19]:

32.5.15 P2 Input Pin State register for EMC pins (P2_INP_STATE - 0x4002 801C)

The P2_INP_STATE register is a read-only register that provides the state of EMC pins EMC_D[31:19]. This allows reading the pin values when the EMC data pins are used as GPIOs P2[12:0].

Table 632. Input Pin State register for EMC pins (P2_INP_STATE - 0x4002 801C)

P2_INP_STATE	Function	Description
0	EMC_D[19] / P2.0	Reflects the state of the P2.0 input pin.
1	EMC_D[20] / P2.1	Reflects the state of the P2.1 input pin.
2	EMC_D[21] / P2.2	Reflects the state of the P2.2 input pin.
3	EMC_D[22] / P2.3	Reflects the state of the P2.3 input pin.
4	EMC_D[23] / P2.4	Reflects the state of the P2.4 input pin.

Het uitlezen van de joystick bij de verschillende acties geeft het volgende resultaat:

1. Geen Actie (0x3FF)
2. Indrukken van de joystick (0x3FE)
3. De overige waarden zijn de overige richtingen

```
reading from 4002801c
address: f402801c, value: 3ff
# echo "r 0x4002801c 1" > /sys/kernel/ES6/hw
sysfile_write (/sys/kernel/ES6/hw) called
buffer: r 0x4002801c 1

reading from 4002801c
address: f402801c, value: 3fe
# echo "r 0x4002801c 1" > /sys/kernel/ES6/hw
sysfile_write (/sys/kernel/ES6/hw) called
buffer: r 0x4002801c 1

reading from 4002801c
address: f402801c, value: 3ed
# echo "r 0x4002801c 1" > /sys/kernel/ES6/hw
sysfile_write (/sys/kernel/ES6/hw) called
buffer: r 0x4002801c 1

reading from 4002801c
address: f402801c, value: 3fb
# echo "r 0x4002801c 1" > /sys/kernel/ES6/hw
sysfile_write (/sys/kernel/ES6/hw) called
buffer: r 0x4002801c 1

reading from 4002801c
address: f402801c, value: 3f7
# echo "r 0x4002801c 1" > /sys/kernel/ES6/hw
sysfile_write (/sys/kernel/ES6/hw) called
buffer: r 0x4002801c 1

reading from 4002801c
address: f402801c, value: 3ff
#
```

Joystick richting het bord bewegen
Joystick niet bewegen

Joystick indrukken

Joystick weg van de pinnetjes bewegen

Joystick richting de pinnetjes bewegen

Joystick weg van het bord bewegen

De joystick richting het bordje bewegen geeft ook een waarde van 0x3FF, dit vonden wij vreemd maar uit overleg is gebleken dat bij meerdere mensen hetzelfde probleem optreed. Het gaat hier om pin P2.1. Uit de datasheet blijkt dat het bewegen van de joystick richting het bordje niet is aangesloten. Zie onderstaande screenshot.

26	P2.1	GPO_12	LCDLE signal. Not used by design . Also connects to ETM pads, if connector mounted.	ETM cannot be used
----	------	--------	--	--------------------

LPC3250_OEM_Board_Users_Guide_Rev_B.pdf - p35

Wat ons opvalt is dat de default waarde 0x3FF is en een bit dus laag wordt als er input wordt gedetecteerd. Daarnaast suggereert 0x3FF dat het om 10 bits gaat terwijl er 12 pinnen zijn.

De onderstaande PDF files zijn gebruikt om de poorten te bepalen op het bordje.

DataSheet-UM10326.pdf (Bitnumbers)

LPC32x0_OEM_Board_v1.1.pdf (Omzetten van LPC -> SODIMM)

Base_schematic_v1.1.pdf (Omzetten van SODIMM -> OEM & Omzetten van OEM -> J)

PORT 0				
BitNr	LPC	SODIMM	OEM	J
0	P0.0	X1-106	P1.8	J3.40
1	P0.1	X1-107	P2.9	J2.24
2	P0.2	X1-31	P2.6	J2.11
3	P0.3	X1-32	P2.7	J2.12
4	P0.4	X1-33	P2.8	J2.13
5	P0.5	X1-34	P2.9	J2.14
6	P0.6	X1-90	P1.22	J3.33
7	P0.7	X1-91	P1.23	J1.27

PORT 1				
BitNr	LPC	SODIMM	OEM	J
0	P1.0			
1	P1.1			
2	P1.2			
3	P1.3			
4	P1.4			
5	P1.5			
6	P1.6			
7	P1.7			
8	P1.8			
9	P1.9			
10	P1.10			
11	P1.11			
12	P1.12			
13	P1.13			
14	P1.14			
15	P1.15			
16	P1.16			
17	P1.17			
18	P1.18			
19	P1.19			
20	P1.20			
21	P1.21			

22	P1.22			
23	P1.23			

Wij gebruiken Port 1 niet voor GPIO omdat uit de datasheet blijkt dat register P1_MUX_SET[23:0] hoog gezet moet worden. Op het moment dat dit gedaan wordt werkt de EMC(external memory control) niet meer. Wij gebruiken de EMC om de USB te lezen waar ons OS op staat. Dus op het moment dat wij GPIO op poort 1 zouden willen gebruiken is het niet mogelijk om het OS op de USB uit te lezen waardoor niks meer zal werken. Zie onderstaand plaatje.

33.4.4 Port 1 multiplexer Set register (P1_MUX_SET - 0x4002 8130)

The P1_MUX_SET register is a write-only register that allows configuring selected pins for one of two functions. In each case, one function is an EMC address and the other is a GPIO-type function.

Writing a one to a bit in P1_MUX_SET results in the corresponding pin being configured as a GPIO function P1[23:0]. Writing a zero to a bit in P1_MUX_SET has no effect.

Table 659. Port 1 Multiplexer Set register (P1_MUX_SET - 0x4002 8130)

P1_MUX_SET	Function	Description
31:24	Reserved	Reserved, user software should not write ones to reserved bits.
23	EMC_A[23] / P1.23	1: Configure as P1.23
22	EMC_A[22] / P1.22	1: Configure as P1.22
21	EMC_A[21] / P1.21	1: Configure as P1.21

PORT 2				
BitNr	LPC	SODIMM	OEM	J
0	P2.0	X1-120	P2.22	J3.47
1	P2.1	X1-121	P2.23	J3.56
2	P2.2	X1-122	P2.25	J3.48
3	P2.3	X1-123	P2.26	J3.57
4	P2.4	X1-124	P2.27	J3.49
5	P2.5	X1-125	P2.30	J3.58
6	P2.6	X1-126	P2.31	J3.50

7	P2.7	X1-116	P2.14	J3.45
8	P2.8	X1-176	P3.27	J1.49
9	P2.9	X1-178	P3.26	J1.50
10	P2.10	X1-180	P3.25	J1.51
11	P2.11	X1-182	P3.24	J1.52
12	P2.12	X1-184	P3.23	J1.53

PORT 3				
BitNr To Set	BitNr To Read	LPC	SODIMM	OEM
25	10	P3.0	X1-117	P2.15
26	11	P3.1	X1-118	P2.19
27	12	P3.2		
28	13	P3.3		
29	14	P3.4	X1-96	P1.28
30	24	P3.5	X1-85	P1.13

Poort 3 heeft een verschil in bitNr op het moment dat er geschreven of gelezen moet worden.

Poort 3 hebben we geconstateerd dat P3.2 en P3.3 Niet connected zijn. Hier zijn we achter gekomen door het op te zoeken in LPC3250_OEM_Board_Users_Guide_Rev_B.pdf. Zie onderstaand screenshot.

<i>Pin</i>	<i>Available on expansion connector</i>
KEY_COL1 / ENET_RX_CLK / ENET_REF_CLK KEY_COL2 / ENET_RX_ER KEY_COL3 / ENET_CR_S KEY_COL4 / ENET_RXD0 KEY_COL5 / ENET_RXD1 KEY_ROW3 / ENET_TX_EN KEY_ROW4 / ENET_TXD0 KEY_ROW5 / ENET_TXD1 GPIO_02 / KEY_ROW6 / ENET_MDC GPIO_03 / KEY_ROW7 / ENET_MDIO	No, connected to on-board Ethernet PHY, DP83848

Verder hebben we een bash file gemaakt (startGPIO) die voor ons:

- De kernel aanroept
- De device aanmaakt
- De devices RW rechten geeft

Dit ziet er zo uit:

```
insmod ESgpio.ko

#Port 0
mknod /dev/J3_40 c 65 40
mknod /dev/J2_24 c 65 224
mknod /dev/J2_11 c 65 211

****

#Read and write rights
chmod 666 /dev/J*
```

Ook hebben we een bash file gemaakt (stopGPIO) die voor ons:

- De kernel stopt
- De devices verwijderd

Dit ziet er zo uit:

```
rmmod ESgpio.ko

#Port 0
rm /dev/J3_40
rm /dev/J2_24

****
```

Op het moment dat de user een device probeert te configureren voor een pin die geen GPIO is, krijgt de user onderstaande foutmelding.

```
# ./startGpio
/sys/kernel/gpio/config created
example to configure J2_24 as output: "echo "J2 24 0" > /sys/kernel/gpio/config"
# ls /dev/
J1_24      J2_11      J3_40      J3_54      pwm1_enable  shm
J1_27      J2_12      J3_45      J3_56      pwm1_freq    tty2
J1_49      J2_13      J3_46      J3_57      pwm2_duty    tty3
J1_50      J2_14      J3_47      J3_58      pwm2_enable  tty4
J1_51      J2_24      J3_48      mtblock3   pwm2_freq
J1_52      J3_33      J3_49      null       sda1
J1_53      J3_36      J3_50      pwm1_duty  sda2
# echo "J1 35 0" > /sys/kernel/gpio/config
sysfile_write (/sys/kernel/gpio/config) called
buffer: J1 35 0

The requested pin is not GPIO
#
```

Indien de gebruiker zelf de device aanmaakt voor een pin op dezelfde manier als wij dat doen:

```
# mknod /dev/J1_35 c 65 135
# ls /dev/
J1_24      J1_53      J3_36      J3_50      pwm1_duty  sda2
J1_27      J2_11      J3_40      J3_54      pwm1_enable shm
J1_35      J2_12      J3_45      J3_56      pwm1_freq  tty2
J1_43      J2_13      J3_46      J3_57      pwm2_duty  tty3
J1_50      J2_14      J3_47      J3_58      pwm2_enable tty4
J1_51      J2_24      J3_48      mtblock3   pwm2_freq
J1_52      J3_33      J3_49      null       sda1
#
```

Krijgt de gebruiker een error als hij deze probeert te gebruiken:

```
# echo 1 > /dev/J1_35
The requested pin is not GPIO
#
```


Testing

Met het commando

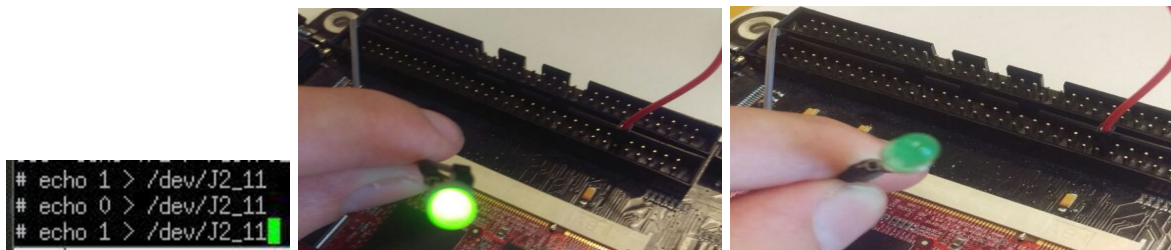
“echo “J2 11 0” > /sys/kernel/gpio/config”

Wordt J2.11 gezet als output, dit is te zien in onderstaand screenshot

```
# echo "J2 11 0" > /sys/kernel/gpio/config sysfile_write (/sys/kernel/gpio/config) called
buffer: J2 11 0

Configured J2_11 as output
Use "echo 0/1 > /dev/J2_11" to write the pin
```

Als we vervolgens “echo 1 > /dev/J2_11” en “echo 0 > /dev/J2_11” kunnen we een lampje aan en uit laten gaan dit is te zien in onderstaande plaatjes.



Vervolgens hebben we de Joystick getest die op pin J3.47 zit. In onderstaand plaatje wordt bij het eerste “cat” commando wordt de joystick ingedrukt en bij het tweede “cat” commando niks gedaan. Hier is dus duidelijk te zien dat we de waarde van deze pin goed uitgelezen wordt.

```
# echo "J3 47 1" > /sys/kernel/gpio/config sysfile_write (/sys/kernel/gpio/config) called
buffer: J3 47 1

Configured J3_47 as input
Use "cat /dev/J3_47" to read the pin
# cat /dev/J3_47
1
# cat /dev/J3_47
0
#
```