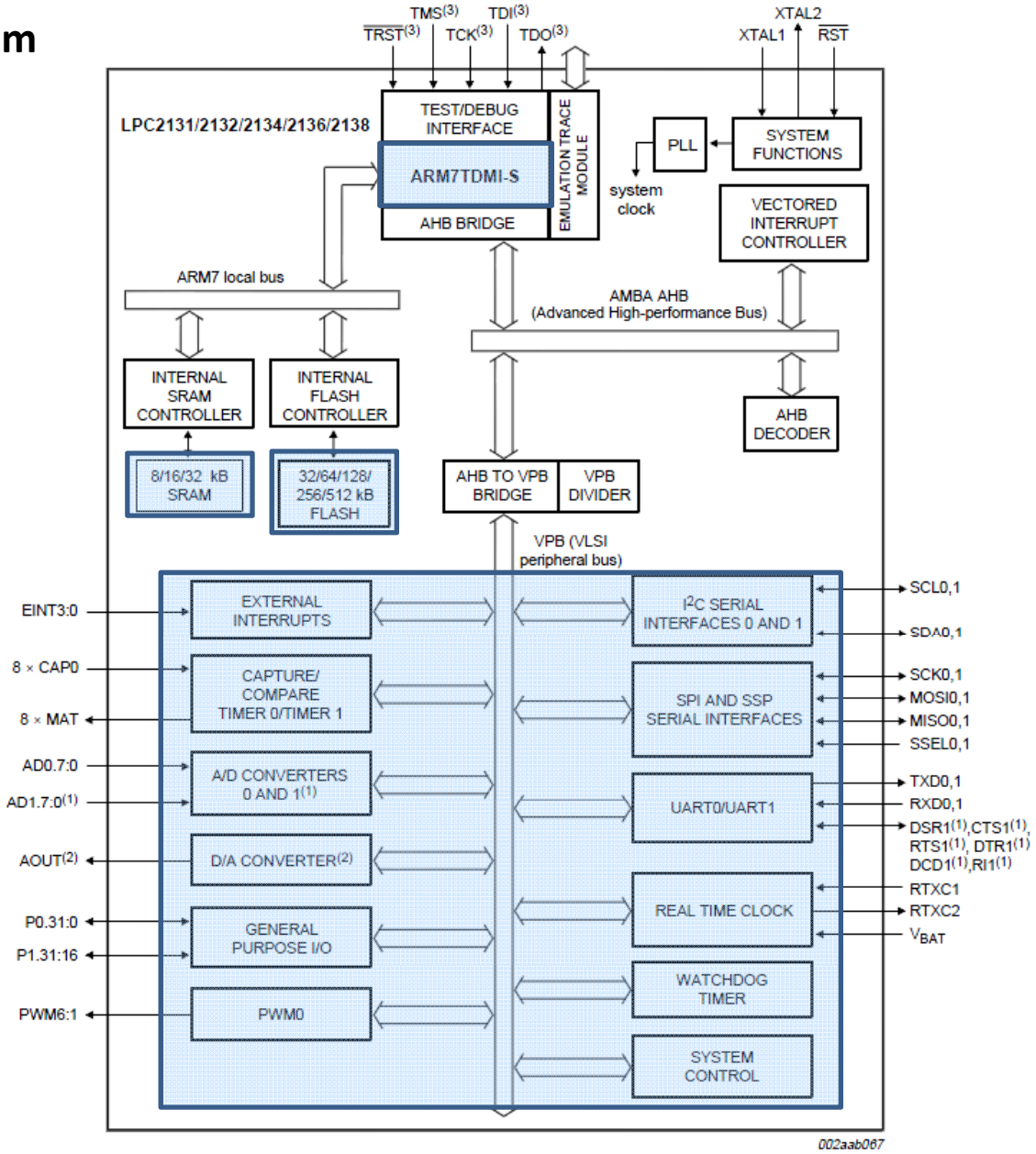
Vaja1
Uvod, programiranje v jeziku C, inicializacija mikrokrmilnika

LPC2138 blok diagram



(1) LPC2134/2136/2138 only.

(2) LPC2132/2134/2136/2138 only.

(3) Pins shared with GPIO.

Fig 1. LPC2131/2/4/6/8 block diagram

Vaja1: Uvod, programiranje v jeziku C, inicializacija mikrokrmilnika

crt0.s:

- **Prepis kode iz FLASH-a v RAM (velikost programa je lahko problem)**
- **Nastavitev prekinitev**
- **Nastavitev skladov za razl. načine delovanja**
- **Inicializacija mikrokrmilnika (MAM, PLL, VPB)**
- **Skok na C funkcijo startup()**

Vaja1:

1. V datoteki startup.c nastavite konfiguracijo pinov tako, da bo pin P0_12 (tipka T0) vhodni in P0_4 (dioda LED0) izhodni

2. Napišite program, ki bo prižgal LED0, ko je tipka T0 pritisnjena in obratno (PAZI: stanje tipk je negirano, vrstni red LED in tipk je obrnjen)

3. Napišite funkcije za delo z GPIO pini

```
- set_pin_direction()  /* inicializacijska funkcija za GPIO pine */
- set_pin_value()      /* zapis logične 0/1 na poljuben pin porta P0 */
- get_pin_value()      /* branje logičnega stanja poljubnega pina porta P0 */
```

4. Dopolnite glavni program, ki bo na LED diodah odražal stanje vseh štirih tipk

- LED: pini P0_4 – P0_7, tipke: pini P0_12 – P0_15

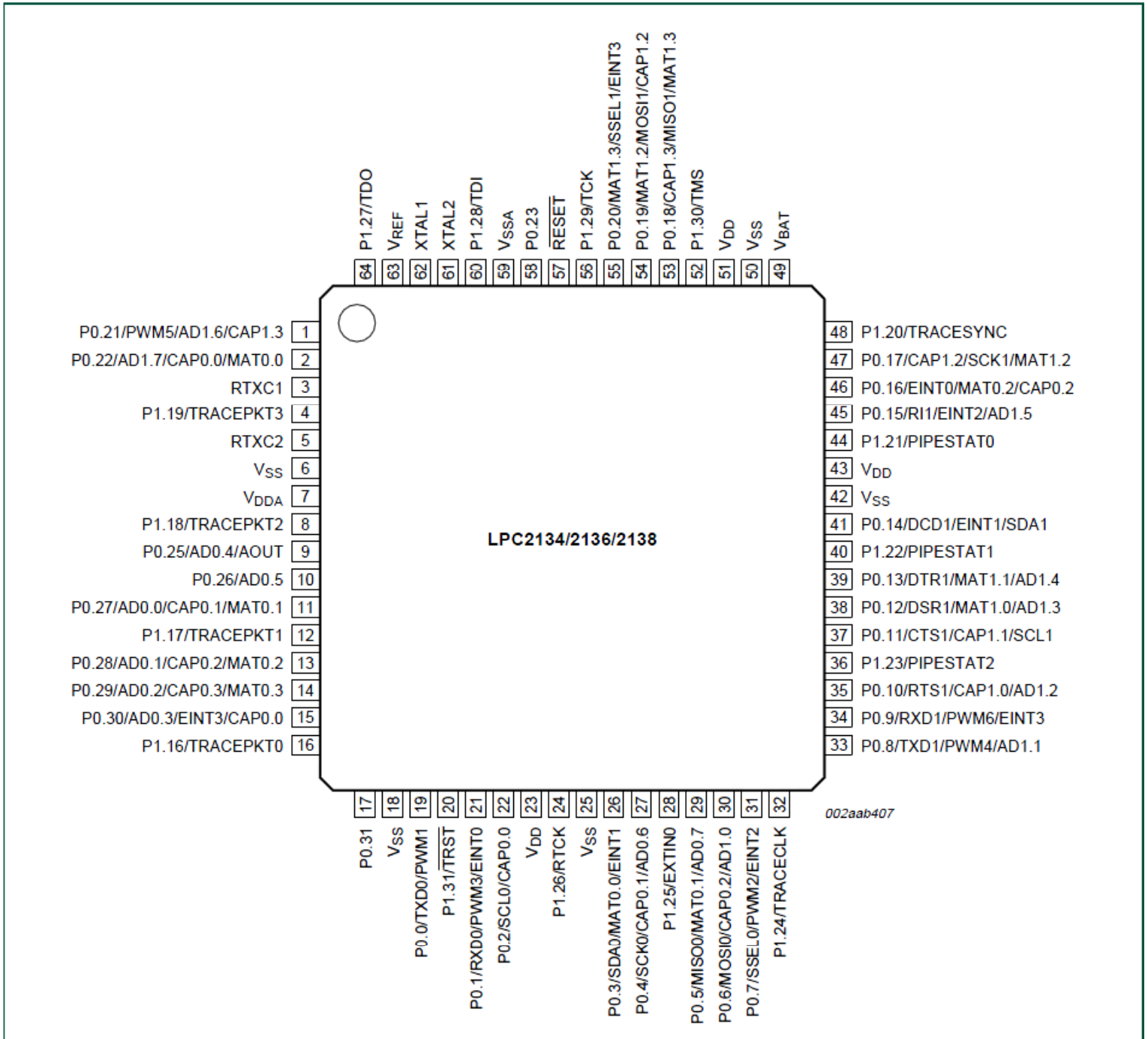


Fig 17. LPC2134/6/8 64-pin package

Table 56: Pin description

Symbol	Pin	Type	Description
P0.0 to P0.31		I/O	**Port 0:** Port 0 is a 32-bit I/O port with individual direction controls for each bit. Total of 30 pins of the Port 0 can be used as a general purpose bi-directional digital I/Os while P0.31 is output only pin. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pin P0.24 is not available.
P0.0/TXD0/ PWM1	19[1]	I/O	**P0.0 —** General purpose digital input/output pin
		O	**TXD0 —** Transmitter output for UART0
		O	**PWM1 —** Pulse Width Modulator output 1
P0.1/RxD0/ PWM3/EINT0	21[2]	I/O	**P0.1 —** General purpose digital input/output pin
		I	**RxD0 —** Receiver input for UART0
		O	**PWM3 —** Pulse Width Modulator output 3
		I	**EINT0 —** External interrupt 0 input
P0.2/SCL0/ CAP0.0	22[3]	I/O	**P0.2 —** General purpose digital input/output pin
		I/O	**SCL0 —** I^2C0 clock input/output. Open drain output (for I^2C compliance)
		I	**CAP0.0 —** Capture input for Timer 0, channel 0
P0.3/SDA0/ MAT0.0/EINT1	26[3]	I/O	**P0.3 —** General purpose digital input/output pin
		I/O	**SDA0 —** I^2C0 data input/output. Open drain output (for I^2C compliance)
		O	**MAT0.0 —** Match output for Timer 0, channel 0
		I	**EINT1 —** External interrupt 1 input
P0.4/SCK0/ CAP0.1/AD0.6	27[4]	I/O	**P0.4 —** General purpose digital input/output pin
		I/O	**SCK0 —** Serial clock for SPI0. SPI clock output from master or input to slave
		I	**CAP0.1 —** Capture input for Timer 0, channel 0
		I	**AD0.6 —** A/D converter 0, input 6. This analog input is always connected to its pin
P0.5/MISO0/ MAT0.1/AD0.7	29[4]	I/O	**P0.5 —** General purpose digital input/output pin
		I/O	**MISO0 —** Master In Slave OUT for SPI0. Data input to SPI master or data output from SPI slave
		O	**MAT0.1 —** Match output for Timer 0, channel 1
		I	**AD0.7 —** A/D converter 0, input 7. This analog input is always connected to its pin

Izbira funkcije pinov:

The Pin Control Module contains 2 registers as shown in Table 57 below.

Table 57: Pin connect block register map

Name	Description	Access	Reset value[1]	Address
PINSEL0	Pin function select register 0.	Read/Write	0x0000 0000	0xE002 C000
PINSEL1	Pin function select register 1.	Read/Write	0x0000 0000	0xE002 C004
PINSEL2	Pin function select register 2.	Read/Write	See Table 60	0xE002 C014

Table 58: Pin function Select register 0 (PINSEL0 - address 0xE002 C000) bit description

Bit	Symbol	Value	Function	Reset value
1:0	P0.0	00	GPIO Port 0.0	0
		01	TXD (UART0)	
		10	PWM1	
		11	Reserved	
3:2	P0.1	00	GPIO Port 0.1	0
		01	RxD (UART0)	
		10	PWM3	
		11	EINT0	
5:4	P0.2	00	GPIO Port 0.2	0
		01	SCL0 (I^2C0)	
		10	Capture 0.0 (Timer 0)	
		11	Reserved	
7:6	P0.3	00	GPIO Port 0.3	0
		01	SDA0 (I^2C0)	
		10	Match 0.0 (Timer 0)	
		11	EINT1	
9:8	P0.4	00	GPIO Port 0.4	0
		01	SCK0 (SPI0)	
		10	Capture 0.1 (Timer 0)	
		11	AD0.6	

Table 63: GPIO register map

Generic Name	Description	Access	Reset value[1]	PORT0 Address & Name	PORT1 Address & Name
IOPIN	GPIO Port Pin value register. The current state of the GPIO configured port pins can always be read from this register, regardless of pin direction.	R/W	NA	0xE002 8000 IO0PIN	0xE002 8010 IO1PIN
IOSET	GPIO Port Output set register. This register controls the state of output pins in conjunction with the IOCLR register. Writing ones produces highs at the corresponding port pins. Writing zeroes has no effect.	R/W	0x0000 0000	0xE002 8004 IO0SET	0xE002 8014 IO1SET
IODIR	GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0x0000 0000	0xE002 8008 IO0DIR	0xE002 8018 IO1DIR
IOCLR	GPIO Port Output clear register. This register controls the state of output pins. Writing ones produces lows at the corresponding port pins and clears the corresponding bits in the IOSET register. Writing zeroes has no effect.	WO	0x0000 0000	0xE002 800C IO0CLR	0xE002 801C IO1CLR

Table 70: GPIO Direction Register 0 (IO0DIR - address 0xE002 8008) bit description

Bit	Symbol	Value	Description	Reset value
31:0	P0xDIR		Direction control bits. Bit 0 in IO0DIR controls P0.0 ... Bit 30 in IO0DIR controls P0.30.	0x0000 0000
		0	Controlled pin is input.	
		1	Controlled pin is output.	

Registri za delo z GPIO:

```
#define IO0PIN      (*((volatile int *)0xe0028000))   // Vrednosti pinov
#define IO0SET      (*((volatile int *)0xe0028004))   // Zapis 1 na pine
#define IO0DIR      (*((volatile int *)0xe0028008))   // Nastavitev smeri pinov
#define IO0CLR      (*((volatile int *)0xe002800c))   // Zapis 0 na pine
#define PINSEL0     (*((volatile int *)0xe002c000))   // Izbira funkcije na pinih P0_0 – P0_14
#define PINSEL1     (*((volatile int *)0xe002c004))  // Izbira funkcije na pinih P0_15 – P0_16
```

Konfiguracija pinov porta P0:

PINSEL0 : xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

Pin0: **če je definiran kot vh./izh. => v PINSEL0 vpišemo kombinacijo 00 na spodnja dva bita (ostale pustimo)**
PINSEL0 <= xxxx xxxx xxxx xxxx xxxx xxxx xxxx xx00

pin_mask = 0000 … 0000 0001 (0x00000001) **: kateri pin nastavljamo**
pinsel_mask = 1111 … 1111 1100 (0xfffffffc) **: maska z kombinacijo 00 na mestih, ki določajo funkcijo pina P0_0**

Pin1:
pin_mask = 0000 … 0000 0010 (0x00000002)
pinsel_mask = 1111 … 1111 0011 (0xfffffff3)
…
Pin15:
pin_mask = 0x00008000
pinsel_mask = 0x3fffffff

Pin16: **=> za pin16-pin31 vpisujemo izbiro funk. v PINSEL1**
pin_mask = 0x00010000
pinsel_mask = 0xfffffffc
…

Nastavitev smeri GPIO pinov:

- **vhodni pini : v pripadajoče bite (pin_mask) v IO0DIR zapišemo 0**
- **izhodni pini : v pripadajoče bite (pin_mask) v IO0DIR zapišemo 1**

/* Zapis vrednosti na izh. pin */

Npr: na pin P0_4 (LED0) želimo vpisati:

- **1 => v IO0SET vpisat 1 na bit4**
- **0 => v IO0CLR vpisat 1 na bit4**

/* Branje vrednost pina */
Npr: pin P0_12 (tipka T0)

Iz registra IO0PIN preberemo vrednost bita 12

GPIO modul:

1. gpio.h: prototipi funkcij in deklaracije spremenljivk za delo z GPIO, naslovi registrov...

```
/* Prototipi funk.: */
void set_pin_direction(int input, int output);    /* izberi funk. in smer pinov */
void set_pin_value(int pin, int value);           /* vpiši vrednost na izh. pin */
int get_pin_value(int pin);                       /* preberi vrednost pina */

/* Argumenti funk.:
*         int input : maska vh. pinov
*         int output : maska izh. pinov
*         int pin : zaporedna št. pina (0 – 31)
*         int value : vrednost, ki jo želimo vpisat na izh. pin (0, 1)
*/
```

2. gpio.c: definicije spremenljivk in funkcij

3. povsod, kjer želimo uporabiti GPIO funkcionalnosti, dodamo #include “gpio.h”