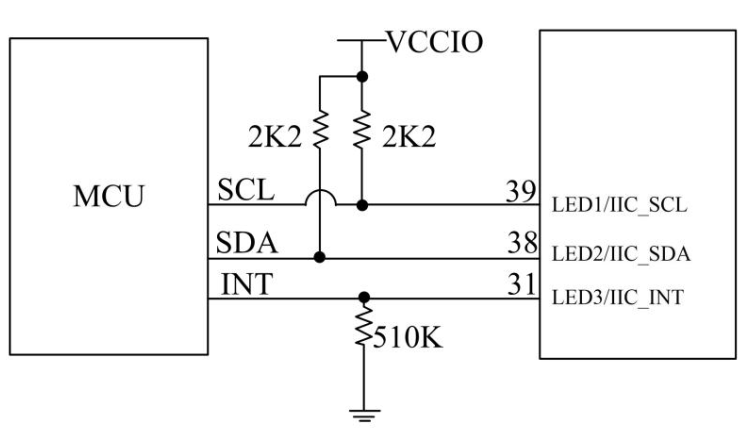
# Typical application description

## I2C connection method

IP2366 can be used as a slave device. The MCU can read or set the voltage, current, power and other information of IP2366 through the I2C interface. The connection method of IP2366 I2C is as follows:



## I2C Precautions

* IP2366 I2C device address: written as 0xEA, read as 0xEB. If you need to set it to another address, you can customize it;
* The I2C communication voltage of IP2366 is 3.3V. If the MCU terminal is 5V voltage, you need to add a level conversion chip to 3.3V;
* IP2366 INT application description: IP2366 will wake up when it detects that INT is high during sleep. After waking up, IP2366 actively pulls up INT. After 100ms, the MCU can perform I2C communication and register read and write operations; Before IP2366 enters sleep, it will switch to the input high resistance to detect the INT state. If it is high, it is considered that the MCU does not allow IP2366 to enter sleep, if it is low, then IP2366 enters sleep; after the MCU detects that INT is low, 16ms to stop accessing the IC;
* The I2C of IP2366 supports a maximum communication frequency of 250k. Considering the clock deviation, it is recommended that the MCU's I2C communication clock use 100k-200k;
* If you want to modify the value of a register in IP2366, you need to read out the value of the corresponding register first, and then after the bit that needs to be modified is summed or calculated, the calculated value is written to the register. Other unopened registers cannot be modified at will. The default value of the register is subject to the read value. The default value of different ICS may be different;
* IP2366 I2C communication is real-time data. After receiving the request, it needs to be interrupted for data preparation. The preparation time is long, so when the MCU communicates with I2C, it needs to determine whether the ACK is received after sending the address and increase the delay by 50us (refer to the I2C application example); it is recommended to read a single byte, 100k I2C communication frequency, and increase the delay by 1ms between each byte;
* At the end of the I2C read data, after the last byte is read, a NACK signal must be given, otherwise IP2366 will think that the data is still being read, and the next clock will continue to output the next data, resulting in the STOP signal cannot be received, and the last read error;
* Reserved registers cannot write data at will, and the original value cannot be changed, otherwise unexpected results will occur. The operation of the register must be carried out in accordance with read-modify-write, only the bits to be used are modified, and the values of other unused bits cannot be modified;
* This document is only for IP2366\_I2C model, other models are invalid;

## I2C application example

After the IP2366 INT pin continues to be high for 100ms, the MCU can perform I2C communication. You can initialize the register first (only modify the register when you need to modify special functions, if you don't need to modify, you cannot write the register); then read the internal information of IP2366 (power, charge and discharge status, button status); and finally perform special operations (such as special indicators, charge and discharge management, fast charge request management); after the MCU detects that the INT is low, it needs to stop accessing I2C within 16ms.

For example:

### Write data 0x5A to 0x05 register

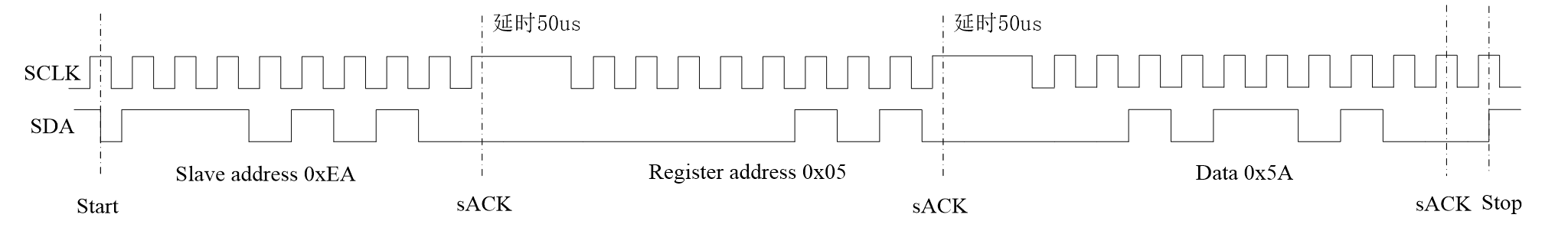


Figure 1: I2C Write 0x05

### Read data from 0x05 register

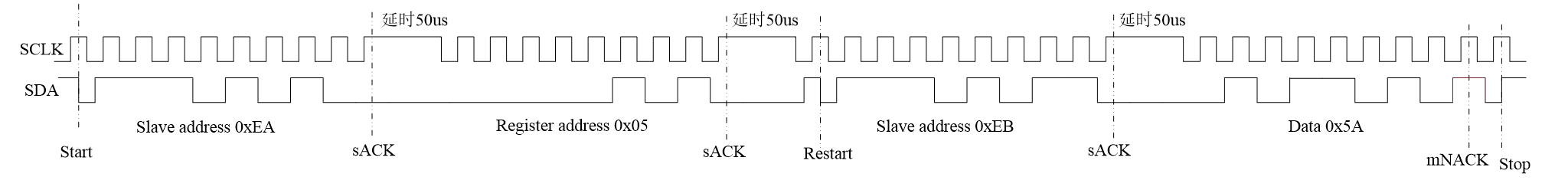


Figure 2: I2C Read 0x05

Actually, read back data from the 0x31 register:

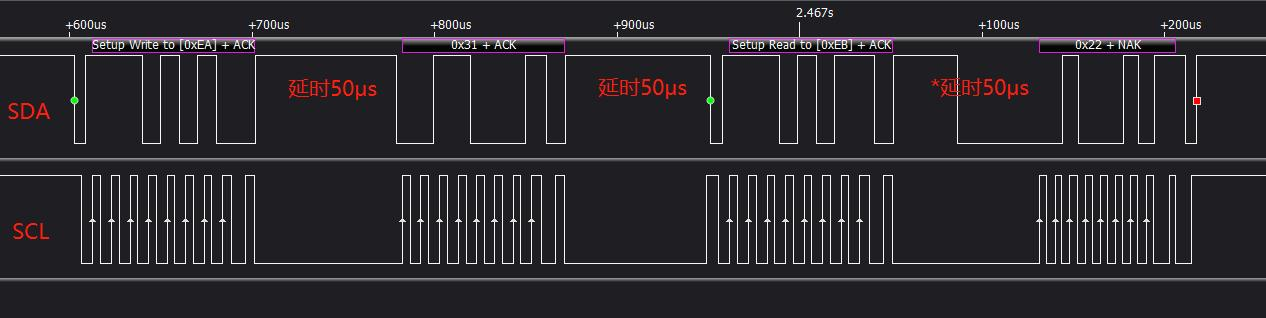


Figure 3: I2C Read 0x31



# Register list:

## Read/write operation register

### [0x00] SYS\_CTL0 (charge enable register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7 | En\_LOADOTP | Enable reset the register value after boot or wake up:  0: Do not reset the register value  1: Reset the register value again  It is not recommended to modify this bit to 0. If it needs to be modified, the software needs to reset the default value of the register regularly, such as after the VINOk VBUOk signal is triggered. | R/W | 1 |
| 6 | En\_RESETMCU | MCU reset register  Write 1: Reset the register to the default value, after reset, the bit will automatically return to 0 | R/W | 0 |
| 5 | En\_INT\_low | When there is an exception, the INT is pulled down by 2MS, indicating that the MCU has an exception.  1: Enable  0: disable | R/W | 0 |
| 4 | En\_Vbus\_SinkDPdM | Type-C port input DM DP fast charge is enabled  1: Enable  0: disable | R/W | 1 |
| 3 | En\_Vbus\_SinkPd | Type-C port input PD fast charge is enabled  1: Enable  0: disable | R/W | 1 |
| 2 | En\_Vbus\_SinkSCP | Type-C port input SCP fast charge is enabled  1: Enable  0: disable | R/W | 1 |
| *1* | *Reserved* |  | *R* | *0* |
| 0 | En\_Charger | Charger Charging is enabled (not configurable after shutdown)  1: Enable  0: disable | R/W | 1 |

### [0x02] SYS\_CTL2 (Vset full charge voltage setting)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:0 | Vset | Full voltage of a single battery  Vset=N\*10+2500mV (up to 4.4V) | R/W | ??? |

### [0x03] SYS\_CTL3 (Iset charging power or current setting)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **description** | **R/w** | **rst** |
| 7:0 | Iset | Maximum battery current limit (cannot be configured to be less than the stop-charge current)  Iset=N\*100mA (maximum 9.7A) | R/W | 0x61 |

### [0x06] SYS\_CTL6 (trickle charging current settings)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:0 | Itk | trickle charging current setting  Itk=N\*50mA | R/W | 0x04 |

### [0x08] SYS\_CTL8 (stop charging current and recharge threshold setting)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:4 | Istop | stop charging current setting  Istop=N\*50mA | R/W | 0x02 |
| 3:2 | Vrch | recharging threshold  00: There is no recharging function after full charge  01: VTRGT-N\*0.05  10: VTRGT-N\*0.1  **11: VTRGT-N\*0.2**  VTRGT–full charge voltage  N–Number of battery cells in series | R/W | 0x02 |
| *1:0* | *Reserved* |  |  |  |

### [0x09] SYS\_CTL9 (standby enable and low voltage setting)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7 | En\_Standby | standby enable  0: disable  1: Enable | R/W | 1 |
| 6 | Standby | Write 1 to enter standby mode, valid once (bit7 must be enabled to write 1)  0: Normal process  1: Enter standby mode immediately when not charging | R/W | 0 |
| 5 | En\_BAT\_Low | 5V low power shutdown enable  0: disable  1: Enable (After enabling, the shutdown voltage is fixed at 5V, with only software protection. And the voltage will also change when the trickle current is converted to constant current) | R/W | 0 |
| *4:0* | *Reserved* |  |  |  |

### [0x0A] SYS\_CTL10 (battery low voltage setting)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:5 | Set\_BATlow | Battery low voltage setting (Trickle current to constant current voltage will change accordingly 2.7V and below only have software low power protection)   |  |  |  |  | | --- | --- | --- | --- | | 000: 2.50V \* N  001: 2.60V \* N | **010: 2.70V \* N**  011: 2.80V \* N | 100: 2.90V \* N  101: 3.00V \* N | 110: 3.10V \* N  111: 3.20V \* N |   N: Number of battery cells in series | R/W | 0x02 |
| *4:0* | *Reserved* |  |  |  |

### [0x0B] SYS\_CTL11 (output enable register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7 | En\_Dc\_Dc\_Output | Discharge output is enabled (not output after shutdown)  1: Enable  0: Do not enable | R/W | 1 |
| 6 | En\_Vbus\_Src\_DP\_dM | C port output DP/DM fast charge is enabled  1: Enable  0: Disable | R/W | 1 |
| 5 | En\_Vbus\_SrcPd | C Port output PD fast charge is enabled  1: Enable  0: disable | R/W | 1 |
| 4 | En\_Vbus\_SrcSCP | C port output SCP fast charge is enabled  1: Enable  0: disable | R/W | 1 |
| *3:0* | *Reserved* |  |  |  |

### [0x0C] SYS\_CTL12 (output maximum power selection register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:5 | Vbus\_Src\_Power | Vbus1 Output power selection:  000:30W  001:45W  010:60W  011:65W  100:100W  **101:140W** | R/W | 0x05 |
| *4:0* | *Reserved* |  |  |  |

More than 60W requires an Emark recognition circuit.

### [0x0D] SELECT\_PDO (select charging PDO gear)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| *7:3* | *Reserved* |  | *R/W* | *0* |
| 2:0 | Pdo\_select | selects the charging PDO mode  000:5V  001:9W  010:12V  011:15V  100:20V | R | ? |

You need to read the corresponding gear of 0x35 before you can select it. The maximum PD gear of the adapter is selected by default.

The configuration becomes invalid after this, and the adapter position needs to be re-identified and reconfigured.

### [0x22] TypeC\_CTL8 (TYPE-C mode control register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:6 | Vbus\_Mode\_Set | Vbus CC mode selection  00: UFP: Upstream Facing Port (device connected as a consumer)  01: DFP: Downstream Facing Port (device connected as a power source)  11: DRP: Dual Role Port (device can act as both a source and a consumer) | R/W | 0 |
| *5:0* | *Reserved* |  |  |  |

### [0x23] TypeC\_CTL9 (output Pdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7 | En\_5VPdo\_3A/2.4 A | 5VPdo current setting  1: 3A  0: 2.4 A | R/W | 1 |
| 6 | En\_Pps2Pdo\_Iset | Pps2 Pdo current set enable  1: Enable  0: disable  \* After enabling, the output power and overcurrent are subject to the set Pdo current, and the overcurrent is 1.1 times the set Pdo current. | R/W | 0 |
| 5 | En\_Pps1Pdo\_Iset | Pps1 Pdo current set enable  1: Enable  0: disable  \* After enabling, the output power and overcurrent are subject to the set Pdo current, and the overcurrent is 1.1 times the set Pdo current. | R/W | 0 |
| 4 | En\_20VPdo\_Iset | 20VPdo current set enable  1: Enable  0: disable  \* After enabling, the output power and overcurrent are subject to the set Pdo current, and the overcurrent is 1.1 times the set Pdo current. | R/W | 0 |
| 3 | En\_15VPdo\_Iset | 15VPdo current set enable  1: Enable  0: disable  \* After enabling, the output power and overcurrent are subject to the set Pdo current, and the overcurrent is 1.1 times the set Pdo current. | R/W | 0 |
| 2 | En\_12VPdo\_Iset | 12VPdo current set enable  1: Enable  0: disable  \* After enabling, the output power and overcurrent are subject to the set Pdo current, and the overcurrent is 1.1 times the set Pdo current. | R/W | 0 |
| 1 | En\_9VPdo\_Iset | 9VPdo current set enable  1: Enable  0: disable  \* After enabling, the output power and overcurrent are subject to the set Pdo current, and the overcurrent is 1.1 times the set Pdo current. | R/W | 0 |
| 0 | En\_5VPdo\_Iset | 5VPdo current set enable  1: Enable  0: disable | R/W | 0 |

### [0x24] TypeC\_CTL10 (5VPdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | 5VPdo\_Iset | 5VPdo current setting  5VPdo=20mA\*N (default 3A,Max=3A) | R/W | 0x96 |

### [0x25] TypeC\_CTL11 (9VPdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | 9VPdo\_Iset | 9VPdo current setting  9VPdo=20mA\*N (default 3A,Max=3A) | R/W | 0x96 |

### [0x26] TypeC\_CTL12 (12VPdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | 12VPdo\_Iset | 12VPdo current setting  12VPdo=20mA\*N (default 3A,Max=3A) | R/W | 0x96 |

### [0x27] TypeC\_CTL13 (15VPdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | 15VPdo\_Iset | 15VPdo current setting  15VPdo=20mA\*N (default 3A,Max=3A) | R/W | 0x96 |

### [0x28] TypeC\_CTL14 (20VPdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | 20VPdo\_Iset | 20VPdo current setting  20VPdo=20mA\*N  (Default is 5A, emark needs to be recognized, Max=5A) if emark is not recognized, 3A | R/W | 0xFA |

### [0x29] TypeC\_CTL23 (Pps1 Pdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | Pps1Pdo\_Iset | Pps1 Pdo current setting  Pps1 Pdo=50mA\*N  (Default is 5A, emark needs to be recognized, Max=5A) if emark is not recognized, 3A | R/W | 0x3C |

### [0x2A] TypeC\_CTL24 (Pps2 Pdo current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** |  | **rst** |
| 7:0 | Pps2Pdo\_Iset | Pps2 Pdo current setting  Pps2 Pdo=50mA\*N  (Default is 5A, emark needs to be recognized, Max=5A) if emark is not recognized, 3A | R/W | 0x3C |

### [0x2B] TypeC\_CTL17 (output Pdo setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| *7* | *Reserved* |  | *R* | *R* |
| 6 | En\_Src\_Pps2Pdo | Pps2 Pdo is enabled  1: Enable  0: disable  \* There is no Pps Pdo after disable | R/W | 1 |
| 5 | En\_Src\_Pps1Pdo  En\_Src\_Pps1Pdo | Pps1 Pdo is enabled  1: Enable  0: disable  \* There is no Pps1 Pdo after disable | R/W | 1 |
| 4 | En\_Src\_20VPdo | 20VPdo enable  1: Enable  0: disable  \* There is no 20V Pdo after disable | R/W | 1 |
| 3 | En\_Src\_15VPdo | 15VPdo enable  1: Enable  0: disable  \* There is no 15V Pdo after disable | R/W | 1 |
| 2 | En\_Src\_12VPdo | 12VPdo enable  1: Enable  0: disable  \* There is no 12V Pdo after disable | R/ | WW 1 |
| 1 | En\_Src\_9VPdo | 9VPdo is enabled  1: Enable  0: disable  \* There is no 9V Pdo after disable | R/W | 1 |
| *0* | *Reserved* |  | *R* | *R* |

### [0x2C] TypeC\_CTL18 (PDO plus 10mA current enable, needs to be configured together with the current setting register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| *7:5* | *Reserved* |  | *R* | *R* |
| 4 | En\_20VPdo\_add | 20VPdo add 10mA current enable  1: Enable  0: disable | R/W | 0 |
| 3 | En\_15VPdo\_add | 15VPdo add 10mA current enable  1: Enable  0: disable | R/W | 0 |
| 2 | En\_12VPdo\_add | 12VPdo add 10mA current enable  1: Enable  0: disable | R/W | 0 |
| 1 | En\_9VPdo\_add | 9VPdo add 10mA current enable  1: Enable  0: disable | R/W | 0 |
| 0 | En\_5VPdo\_add | 5VPdo add 10mA current enable  1: Enable  0: disable | R/W | 0 |

## Read-only status indication register

A must-read section!!!

Multiple registers represent the same state. Each read of the lower 8-bit register will update the upper 8-bit and lower 8-bit data.

To read the register, you must read the lower 8 bits first and then the upper 8 bits to ensure that the same data is read.

The order of the two registers of **BAT** terminal voltage should be to read **0x50** first and then **0x51**.

### [0x31] STATE\_CTL0 (state of charge control register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| *7: 6* | *Reserved* |  | *R* | *X* |
| 5 | CHG\_En | charging flag  1: Charging state (VbusOk means charging state)  0: Non-charging state | R | X |
| 4 | CHG\_End | full state flag  1: fully charged  0: charged not fully | R | X |
| 3 | Output\_En | discharge state Flag  1: Discharge state and the output port has been opened, there is no abnormality  0: Discharge state, the output is not turned on or there is an abnormal discharge | R | X |
| 2:0 | Chg\_state | Chg state  000: Standby  001: Trickle  010: Constant current charging  011: Constant voltage charging  100: Charging is waiting (including when charging is not turned on, etc.)  101: Full state  110: Charging timeout | R | X |

### [0x32] STATE\_CTL1 (state of charge control register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7:6 | Chg\_State | Chg state  00: 5V input charging  01: High voltage input fast charging | R | X |
| *5:0* | *Reserved* |  | *R* | *X* |

### [0x33] STATE\_CTL2 (input Pd status control register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7 | Vbus\_Ok | Vbus Ok  1: Vbus has power  0: Vbus has no power | R | X |
| 6 | Vbus\_Ov | Vbus Ov  1: Vbus input overvoltage  0: Vbus input without overvoltage | R | X |
| *5:3* | *Reserved* |  |  | *X* |
| 2:0 | Chg\_Vbus | charging voltage  111: 20V Charging  110: 15V charging  101: 12V charging  100: 9V charging  011: 7V charging  010: 5V charging | R | X |

### [0x34] TypeC\_STATE (system status indication register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7 | Sink\_Ok | Type-C Sink Input connection flag  1: Valid  0: Invalid | R | X |
| 6 | Src\_Ok | Type-C Src output connection flag  1: Valid  0: Invalid | R | X |
| 5 | Src\_Pd\_Ok | Src\_Pd\_Ok Output connection flag  1: Valid  0: Invalid | R | X |
| 4 | Sink\_Pd\_Ok | Sink Pd Ok input connection flag  1: Valid  0: Invalid | R | X |
| 3 | Vbus\_Sink\_Qc\_Ok | input fast charge valid flag  Qc5Vand Pd5V are not counted as fast charge Ok  1: Valid  0: Invalid | R | X |
| 2 | Vbus\_Src\_Qc\_Ok | output fast charge valid flag  Qc5V and Pd5V are not counted as fast charge Ok  1: Valid  0: Invalid | R | X |
| *1:0* | *Reserved* |  |  |  |

### [0x35] RECEIVED\_PDO (receive PDO gear)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| *7:5* | *Reserved* |  | *R* | *R* |
| 4 | PDO\_20V | The device receives 20VPdo  1: True  0: False | R | X |
| 3 | PDO\_15V | The device receives 15VPdo  1: True  0: False | R | X |
| 2 | PDO\_12V | The device receives 12VPdo  1: True  0: False | R | X |
| 1 | PDO\_9V | The device receives 9VPdo  1: True  0: False | R | X |
| 0 | PDO\_5V | The device receives 5VPdo  1: True  0: False | R | X |

### [0x38] STATE\_CTL3 (system overcurrent indication register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| *7:6* | *Reserved* |  | *R* | *X* |
| 5 | Vsys\_Oc | Vsys outputs the overcurrent flag, you need to write 1 to clear 0  1: Vsys output has a trigger overcurrent signal  0: Vsys output does not trigger overcurrent signal  The system continuously detects more than two overcurrent states within 600mS and considers the overcurrent to be valid, and positions this flag at 1. The external master can read this flag to determine whether there is an abnormal overcurrent; from the occurrence of the overcurrent state to the system sleep, the time is about 1.5s, ~~and this flag will remain at 1 after sleep, so the external master needs to read the flag within this time and make corresponding processing, and then write 1 to clear the flag to 0; if you need to judge that the overcurrent state is revoked, you need to reopen the output port (register 0x22[7] First write 0 Write 1 again), and then read the status flag.~~ | R | X |
| 4 | Vsys\_Scdt | Vsys output short circuit flag, need to write 1 to clear 0  1: Vsys output has a trigger short circuit signal  0: Vsys output does not trigger a short circuit signal  The system continuously detects more than two short-circuit states within 600mS and considers the short-circuit to be valid, and positions this flag at 1. The external master can read this flag to determine whether there is an abnormal short-circuit; from the short-circuit state to the system sleep, the time is about 1.5s, ~~after sleep, this flag will be maintained at 1, so the external master needs to read the flag within this time and make corresponding processing, and then write 1 to clear the flag to 0; if you need to judge that the short-circuit state is revoked, you need to reopen the output port (register 0x22[7] first write 0 and then write 1), Then read the status flag.~~ | R | X |
| *3:0* | *Reserved* |  | *R* |  |

### [0x50] BATVADC\_DAT0 (VBAT voltage register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | BATVADC[7:0] | Low 8bit of BATVADC data  Voltage of VBATPIN | R | X |

### [0x51] BATVADC\_DAT1 (VBAT voltage register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | BATVADC[15:8] | High 8bit of BATVADC data  Voltage of VBATPIN  VBAT=BATVADC (mV) | R | X |

### [0x52] VsysVADC\_DAT0 (Vsys voltage register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | VsysVADC[7:0] | Low 8bit Vsys voltage data  Voltage of VsysPIN | R | X |

### [0x53] VsysVADC\_DAT1 (Vsys voltage register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | VsysVADC[15:8] | Vsys voltage data high 8bit  VsysPIN voltage  Vsys=VsysVADC (mV) | R | X |

### [0x69] TIMENODE1 (the first bit of the timestamp register) (the timestamp symbol is an ASCII character)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | TimeNode1 | The first ASCII symbol corresponds to the value | R | X |

### [0x6A] TIMENODE2 (the second bit of the timestamp register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | TimeNode2 | The second ASCII symbol corresponds to the value | R | X |

### [0x6B] TIMENODE3 (the third bit of the timestamp register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | TimeNode3 | The third ASCII symbol corresponds to the value | R | X |

### [0x6C] TIMENODE4 (the fourth bit of the timestamp register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | TimeNode4 | The fourth ASCII symbol corresponds to the value | R | X |

### [0x6D] TIMENODE5 (the fifth bit of the timestamp register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | TimeNode5 | The fifth ASCII symbol corresponds to the value | R | X |

### [0x6E] IBATIADC\_DAT0 (BAT terminal current register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | IBATIADC[7: 0] | Battery terminal current Low 8bit | R |  |

### [0x6F] IBATIADC\_DAT1 (BAT terminal current register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | IBATIADC[15:8] | Cell terminal current  high 8bits of BATIADC data  IBAT=IBATIADC(mA) | R | X |

### [0x70] ISYS\_IADC\_DAT0 (IVsys terminal current register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | ISYSIADC[7:0] | IVsys terminal current  VsysIADCV data low 8bit | R | X |

### [0x71] IVsys\_IADC\_DAT1 (IVsys terminal current register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | IVsysIADC[15:8] | IVsys terminal current  High 8bits of VsysIADCV data  IVsys= IVsysIADC (mA) | R | X |

### [0x74] Vsys\_POW\_DAT0 (Vsys terminal power register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | Vsys\_POW\_ADC[7:0] | Vsys terminal power ADC data low 8bit | R | X |

### [0x75] Vsys\_POW\_DAT1 (Vsys terminal power register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | Vsys\_POW\_ADC[15:8] | Vsys terminal power ADC data in 8bits | R | X |

### [0x77] INTC\_IADC\_DAT0 (NTC output current register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/w** | **rst** |
| 7 | NTC\_IADC\_DAT | 0: Output 20uA  1: output 80uA | R | X |
| *6:0* | *Reserved* |  |  | *X* |

### [0x78] VGPIO0\_NTC\_DAT0 (VGPIO0\_NTC\_ADC voltage register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | VGPIO0\_DAT0[7:0] | VGPIO0 ADC data low 8bit | R | X |

### [0x79] VGPIO0\_NTC\_DAT1 (VGPIO0\_NTC\_ADC voltage register)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bit(s)** | **Name** | **Description** | **R/W** | **rst** |
| 7:0 | VGPIO0\_DAT1  [15:8] | High0 ADC8High 8bits  of VGPIO 0 ADC data0 VGPIO0 DAT=VGPIO0 ADC (mV)(0~3.3V) | R | X |

# Version/revision history:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **date** | **Revision content** | **Draftsman/**  **revision person** |
| V1.00 | 2023-3-24 | Modify standby EN wake-up; | IT555 |
| V1.10 | 2023-4-18 | Added option to charge PDO Gear | IT555 |
| V1.11 | 2023-4-24 | Delete I2C power reading function description (Read is not supported | IT555 |
| V1.12 | 2023-6-13 | Added standby charging Battery Level Power Wake-up Instructions | IT555 |
| V1.13 | 2023-6-26 | Low voltage gear Low support to 2.5V | IT555 |
|  |  |  |  |