KAT IIC Controller Register Map

Name	Offset	Length	Function
OPFIFO	0x2	0x2	Setting Bit 10 will terminate the IIC byte access with a STOP Sequence and send a “No Acknowledge” with on the associated read. Setting Bit 9 will initiate the IIC byte access with a START Sequence. When bit 8 is high the byte transfer is a read, when low a write. Bits 7:0 contain the data to be written and are unused in the write case This FIFO will latch on a LSByte write.
RXFIFO	0x6	0x2	Bits 7:0 of this FIFO contains the data read during a byte read transfer. This FIFO will latch on a LSByte read.
STATUS	0x8	0x4	Bit[0] RXFIFO empty flag Bit[1] RXFIFO full flag Bit[2] RXFIFO overflow error latch Bit[4] OPFIFO empty flag Bit[5] OPFIFO full flag Bit[6] OPFIFO overflow error latch Bit[8] NACK on write error latch Any write to these register will clear all latches and reset all fifos
CTRL	0xc	0x4	Setting Bit[0] will lock the op fifo, allowing multiple operations to be queued. This is useful in high latency environment where IIC devices are likely to timeout. Clearing bit[0] fill unlock the fifo, allowing operations to commence.

KAT IIC Notes

OPFIFO and RXFIFO are 32 deep.

KAT ADC / KAT IIC Examples (From UBoot)

Note: when entering Uboot commands ensure operations are entered all at once otherwise iic device will timeout

To Access Local (Ambient) Temperature on TMP421 (IIC ADDR 0x4C, REGISTER ADDR 0x0)

```
mw d0040000 298; # START, WRITE, DATA=0x98 (4C << 1 + 0x0(IIC WRITE))
mw d0040000 000; # WRITE, DATA=0x00 (Register Address)
mw d0040000 299; # START(REPEATED), WRITE, DATA=0x99 (4C << 1 + 0x1(IIC READ))
mw d0040000 500; # STOP, READ
md d0040004 1; # Read Data from RXFIFO
```

```
mw d0040000 298; mw d0040000 000; mw d0040000 299; mw d0040000 500; md d0040004 1;
```

To Access ADC Temperature on TMP421 (IIC ADDR 0x4C, REGISTER ADDR 0x1)

```
mw d0040000 298; # START, WRITE, DATA=0x98 (4C << 1 + 0x0(IIC WRITE))
mw d0040000 001; # WRITE, DATA=0x01 (Register Address)
mw d0040000 299; # START(REPEATED), WRITE, DATA=0x99 (4C << 1 + 0x1(IIC READ))
mw d0040000 500; # STOP, READ
md d0040004 1; # Read Data from RXFIFO
```

```
mw d0040000 298; mw d0040000 001; mw d0040000 299; mw d0040000 500; md d0040004 1;
```

To Write EEPROM DATA (IIC ADDR 0x51, ADDR 0x4, DATA 0xDEADBEEF)

```
mw d0040000 2A2; # START, WRITE, DATA=0xA2 (0x51 << 1 + 0x0(IIC WRITE))
mw d0040000 004; # WRITE, DATA=0x04 (Register Address)
mw d0040000 0de; # WRITE, DATA=0xde
mw d0040000 0ad; # WRITE, DATA=0xad
mw d0040000 0be; # WRITE, DATA=0xbe
mw d0040000 4ef; # WRITE, DATA=0xef
```

```
mw d0040000 2A2; mw d0040000 004; mw d0040000 0de; mw d0040000 0ad; mw d0040000 0be;  mw d0040000 4ef;
```

To READ EEPROM DATA (IIC ADDR 0x51, ADDR 0x4)

```
mw d0040000 2A2; # START, WRITE, DATA=0xA2 (0x51 << 1 + 0x0(IIC WRITE))
mw d0040000 004; # WRITE, DATA=0x04 (Register Address)
mw d0040000 2A3; # START(REP), WRITE, DATA=0xA2 (0x51 << 1 + 0x1(IIC READ))
mw d0040000 100; # READ
mw d0040000 100; # READ
mw d0040000 100; # READ
mw d0040000 500; # STOP, READ
md d0040004  1;
md d0040004  1;
md d0040004  1;
md d0040004  1;
```

To SET ADC1 PCA9555 Output 0 [7:0] to enabled (IIC ADDR 0x21, ADDR 0x6)

```
mw d0040000 242; # START, WRITE, DATA=0x42 (21 << 1 + 0x0(IIC WRITE))
mw d0040000 006; # WRITE, DATA=0x06 (Register Address)
mw d0040000 400; # STOP, WRITE, DATA=0x00 (OE_n [7:0] = 0x0)
```

To SET ADC1 PCA9555 Output 0 [7:0] (IIC ADDR 0x21, ADDR 0x2, Data 0xF0)

```
mw d0040000 242; # START, WRITE, DATA=0x42 (21 << 1 + 0x0(IIC WRITE))
mw d0040000 002; # WRITE, DATA=0x02 (Register Address)
mw d0040000 4F0; # STOP, WRITE, DATA=0xF0
```