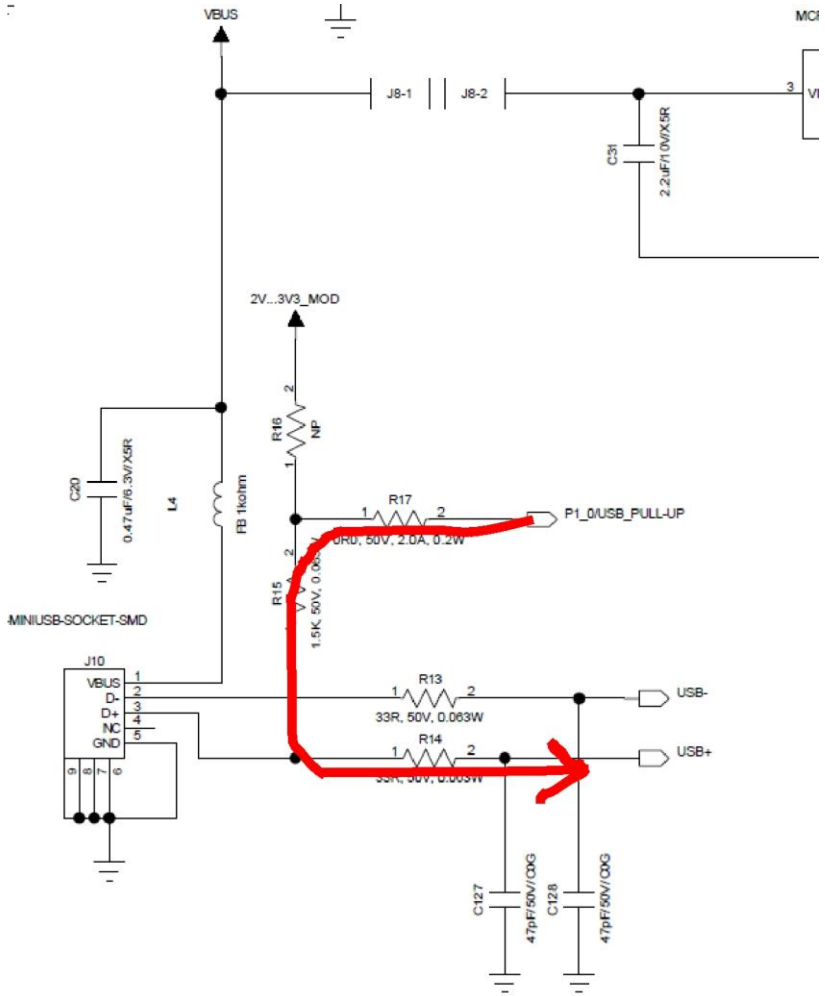
Current Leakage Paths in DKBLE112 – Minimizing the Current Consumption



If the P1_0 is low, then there will be current leakage path from USB+ pin. To avoid excessive leakage with the evaluation board, P1_0 has to be set high. This can be done either by using external pull-up resistor or by configuring P1_0 as output driving high.

- P1_0 and P1_1 pins do not have pull-up or pull-down capability. In DKBLE112 P1_0 has to be set as output driving high and P1_1 has to be set output driving either high or low
- IO’s configured as peripherals do not have pull-up / pull-down capability. Usually the sensors connected to these IO’s will drive the input to a known state. This however will not be the case if the sensors are disconnected with the switch.

The following examples can be used on the evaluation kit to demonstrate power mode 2 (0.9 -1.2uA)

Hardware.xml

```
<hardware>
	<sleeposc enable="true" ppm="30" />
	<port index="0" tristatemask="0" pull="up" />
	<port index="1" tristatemask="0" pull="up" />
	<port index="2" tristatemask="0" pull="up" />
	<pmux regulator_pin="7" />
	<usb enable="false" endpoint="none" />
	<txpower power="15" bias="5" />
	<script enable="true" />
</hardware>
```

BGScript.bgs

```
event system_boot(major,minor,patch,build,ll_version,protocol,hw)

        call hardware_io_port_config_direction(0, $0)
        call hardware_io_port_write(0, $FF, 1)
        call hardware_io_port_config_direction(1, $3)
        call hardware_io_port_write(1, $3, 1)
        call hardware_io_port_config_direction(2, $0)
        call hardware_io_port_write(2, $FF, 1)

        call gap_set_adv_parameters(16000, 16000, 1)

        call gap_set_mode(gap_general_discoverable,gap_undirected_connectable)

end
```