# BLE

To start with BLE we will begin by looking up the chip on the board

## ISP1507-AX

To really understand what this is we look in the datasheets. On figure 1 we can clearly see that the chip contains its own processor.

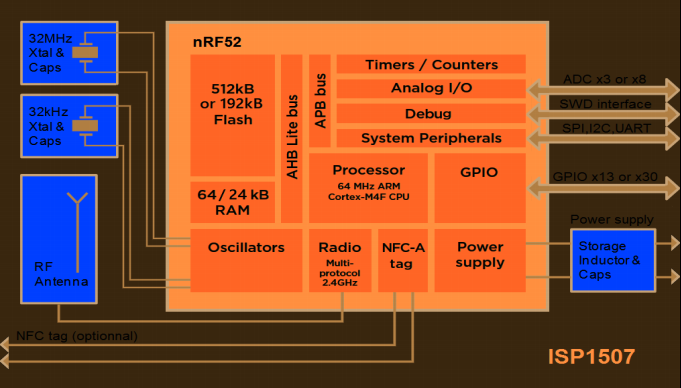
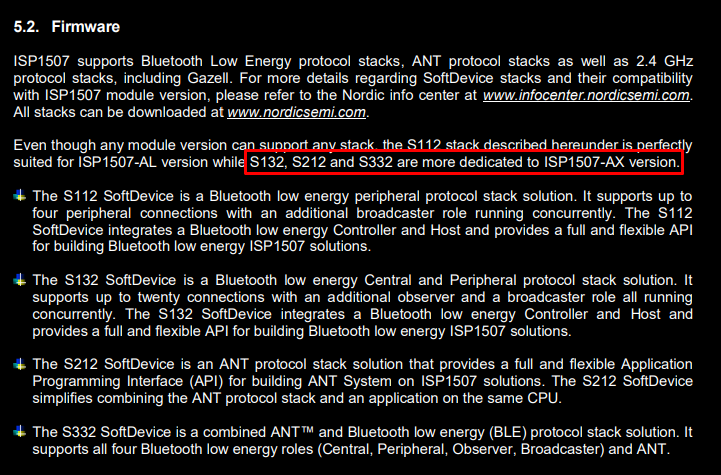


Figure 1. Block Diagram ISP1507-AX

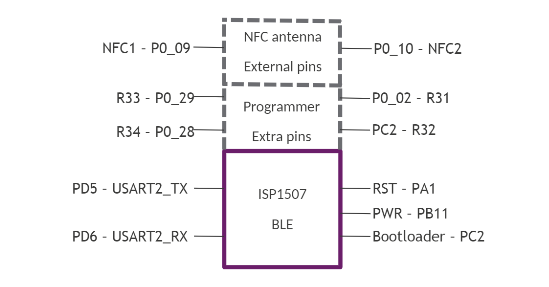
The stack to support this chip has already been written. nRF52 is a very commonly used chip.

Choosing on of these we can program our device.

We take for example S132 because we have the AX version. S132 is a softdevice containing the premade stack. To further understand it read the section on softdevices.

## Verbinden

Om de chip te programmeren zijn er meerdere lijnen voorzien:



We hebben programmer pins gekregen en we kunnen communiceren over USART.

## Softdevices

So a softdevice is essentially a precompiled hex file. This means its programmed separately and has no link time dependency (meaning you don’t have to compile it every time with your application).

These contain: The protocol stacks (S112, S132,…) and they keep the stack and application separate.

So it provides is with an API and we don’t care about the stack anymore. Only interfacing with the API.

The softdevice restricts your access to peripherals inside the ISP1507 chip! To find the list of blocked devices use the Softdevice specifications (manual -> ask satish).

### SoftDevice API’s

We hebben twee soorten API’s.

1. Protocol API’s -> acces voor RF protocol functionality
2. nRF API’s -> System on Chip(SoC) restricted sources (check manual)