**AT vs. API Operating Mode**

|  |  |  |
| --- | --- | --- |
|  | **AT Mode**  **(Transparent Mode)** | **API Mode** |
| **Advantages** | * Provides simple interface for easy XBee startup * Easily supported by applications: Every module gets exactly what you send * Very good for two-way XBee communication | * Can write to or read XBee configuration through network * Can transmit data packets to one or multiple XBee devices, faster than direct connection used in AT mode * Received data includes sending device’s address * Received data includes transmission and reason for success/fail details * Advanced feature: Network Diagnostic and Firmware Upgrades |
| **Disadvantages** | * Cannot write to or read XBee configurations through network * Must update configuration to establish new destinations to transmit data * No way to identify the source of received data (Packets do not include sender’s address) * Received data does not include reason for success/fail or transmission details * Does not offer Network Diagnostics or Firmware Upgrades | * Complex interface: data packets have a very specific format and must be followed exactly * Difficult to be supported by applications due to complexity of data packets. (Data must be parsed to receive data or created to transmit data) * Sent and Received data are not the same; Received data include extra information including control data. |

The XBee S2C modules in the S.M.A.R.T. Alarm system will use the API operating mode. API Mode has been chosen because it has the ability for received packets to include the sending device’s address. The S.M.A.R.T. Alarm system requires this ability so the main hub can identify which smoke alarm has detected a fire to create a path to exit the building. Without this feature, the S.M.A.R.T. Alarm system would be just like any other fire alarm system.

https://www.digi.com/resources/documentation/digidocs/pdfs/90001942-13.pdf