Request Frame มีความยาวทั้งหมด 8 ไบต์ (Bytes) โดยมีรายละเอียดหน้าที่ของแต่ละไบต์เรียงตามลำดับดังนี้

📊 ตารางอธิบายโครงสร้างข้อมูลที่ส่ง (Request Frame)

ไบต์ที่	ข้อมูล (Hex)	ชื่อโครงสร้าง (Field Name)	คำอธิบายหน้าที่การทำงาน
Byte 0	`0x02`	**Slave Address**	หมายเลขอุปกรณ์ (ID) ของเซนเซอร์ตัวที่ต้องการคุยด้วย (ในโค้ดคือตัวที่ 2)
Byte 1	`0x04`	**Function Code**	คำสั่งให้เซนเซอร์ทำอะไร (`0x04` คือสั่งอ่านค่าจาก Input Register)
Byte 2	`0x00`	**Starting Address High**	บิตบนของตำแหน่งเริ่มต้นที่จะอ่าน
Byte 3	`0x01`	**Starting Address Low**	บิตล่างของตำแหน่งเริ่มต้น (`0x0001` คือเริ่มอ่านที่ค่าอุณหภูมิ)
Byte 4	`0x00`	**Quantity of Registers High**	บิตบนของจำนวน Register ที่ต้องการอ่านทั้งหมด
Byte 5	`0x01` / `0x02`	**Quantity of Registers Low**	บิตล่างของจำนวน Register ที่ต้องการอ่าน (`0x01` = 1 ช่อง, `0x02` = 2 ช่อง)
Byte 6	ตัวแปรตามค่าคำนวณ	**CRC Checksum Low**	รหัสตรวจสอบความถูกต้องส่วนท้าย บิตต่ำ (ยึดตามผลคำนวณไบต์ 0-5)
Byte 7	ตัวแปรตามค่าคำนวณ	**CRC Checksum High**	รหัสตรวจสอบความถูกต้องส่วนท้าย บิตสูง (ยึดตามผลคำนวณไบต์ 0-5)

💡 เจาะลึกตำแหน่ง Register ของเซนเซอร์ XY-MD02

อ้างอิงตามคู่มือของเซนเซอร์รุ่นนี้ ค่าเริ่มต้นที่มันเปิดให้เข้าถึงได้จะมีตำแหน่งดังนี้:

- **`0x0001` (Register 1):** เก็บค่าอุณหภูมิ (Temperature)
- **`0x0002` (Register 2):** เก็บค่าความชื้น (Humidity)

ข้อแตกต่างของการเลือกจำนวน Register ใน Byte 5:

- หากส่ง `0x01` เซนเซอร์จะตอบกลับมาเฉพาะค่า **อุณหภูมิ** อย่างเดียว (รวม 7 ไบต์)
- หากส่ง `0x02` เซนเซอร์จะตอบกลับมาทั้งค่า **อุณหภูมิและความชื้น** พร้อมกันในครั้งเดียว (รวม 9 ไบต์) ซึ่งประหยัดเวลากว่าในการดึงข้อมูล

โครงสร้างของข้อมูลที่ตอบกลับ (Response Frame)

เมื่อเราขออ่าน 2 Registers เซนเซอร์จะตอบกลับมาทั้งหมด 9 Bytes ดังนี้:

- `Byte 0` : Address ของอุปกรณ์ (`0x02`)
- `Byte 1` : Function Code (`0x04`)
- `Byte 2` : Byte Count หรือจำนวนบิตข้อมูลที่ส่งกลับมา (`0x04` เพราะมี 2 Registers)
- `Byte 3-4` : ค่าอุณหภูมิ (Temperature High/Low)
- `Byte 5-6` : ค่าความชื้น (Humidity High/Low)
- `Byte 7-8` : ค่า CRC Checksum สำหรับตรวจสอบความถูกต้อง [6, 7]

หากต้องการให้ระบบทำงานได้เสถียรและยืดหยุ่นยิ่งขึ้น คุณอยากให้ผมช่วยเขียนฟังก์ชัน **คำนวณค่า CRC อัตโนมัติ** เพิ่มเติมด้วยไหมครับ? หรือหากคุณพบปัญหา **ค่าที่อ่านออกมาเพี้ยน** สามารถแจ้งเพื่อตรวจสอบวงจรฮาร์ดแวร์ร่วมกันได้เลยครับ [8, 9]