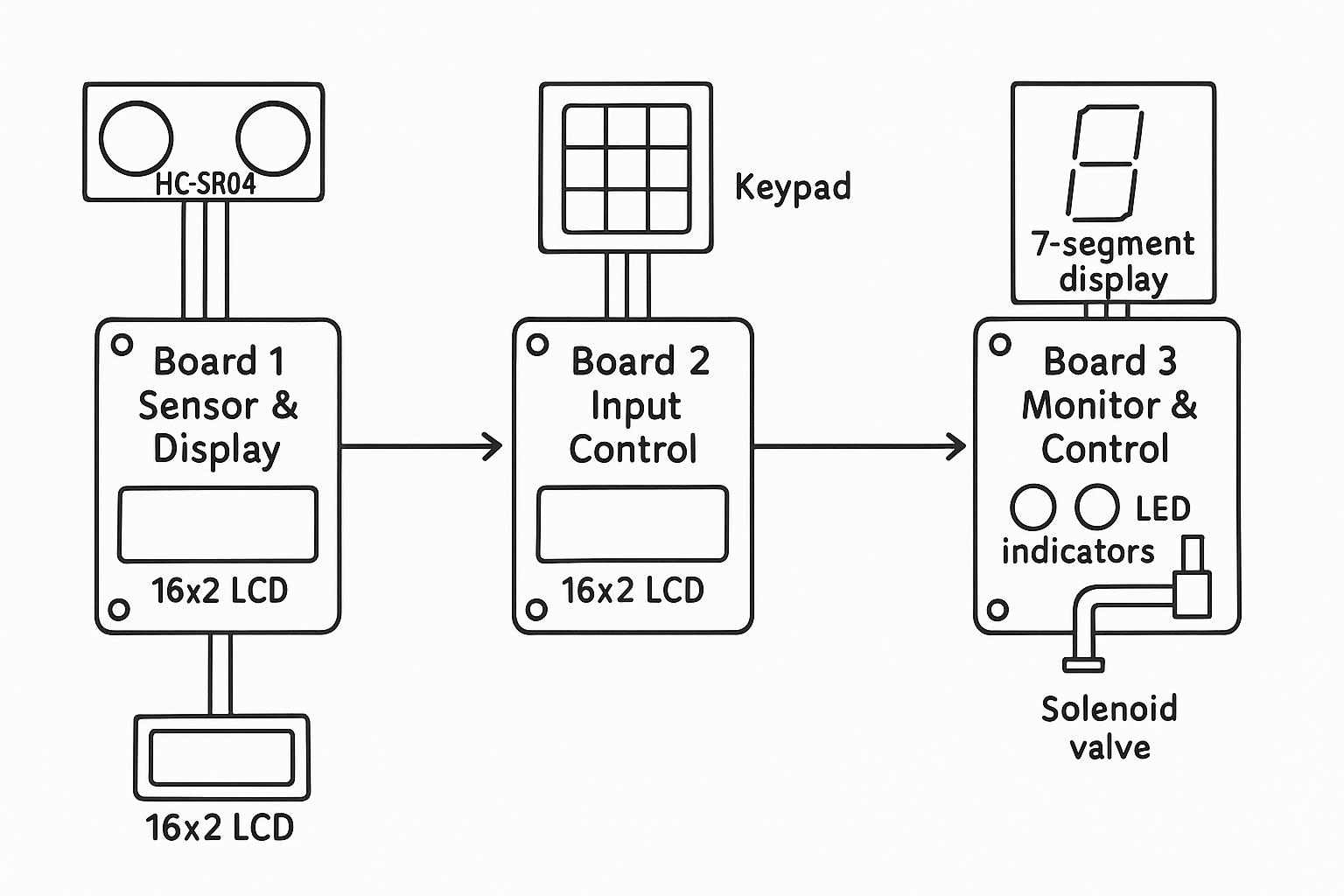
# ระบบควบคุมโรงงานผลิตยาอัตโนมัติ

เอกสารนี้สรุปผลงานจากโจทย์ปฏิบัติ **ระบบควบคุมสายการผลิตยา** ซึ่งแบ่งงานออกเป็น 3 บอร์ดหลักและ 6 บล็อกการทำงาน ตามที่ได้พัฒนาและทดสอบไว้ในโปรเจ็กต์นี้ โค้ดทั้งหมดเขียนด้วยภาษา C แนว **AVR register level** (ไม่ใช้ Arduino library) เพื่อให้สอดคล้องกับวิชาที่เรียน และมีคอมเมนต์ภาษาไทยอธิบายแต่ละส่วนอย่างละเอียด

## ภาพรวมระบบ

ระบบถูกแบ่งออกเป็นบอร์ดย่อย 3 ตัว โดยแต่ละบอร์ดรับผิดชอบหน้าที่ต่างกันและสื่อสารกันผ่าน USART (9600 bps, 8 data bits, Even parity, 1 stop bit) ดังผังด้านล่าง



1. **บอร์ด 1: Sensor & Display** – นับจำนวนเม็ดยาด้วยเซ็นเซอร์ HC‑SR04 และแสดงผลบน LCD 16×2 พร้อมส่งข้อมูลไปบอร์ด 3 และรับคำสั่งจากบอร์ด 2
2. **บอร์ด 2: Input Control** – รับค่าเป้าหมายผ่าน Keypad ADC 4×4, แสดงผลบน LCD 16×2 และส่งคำสั่ง TARGET ให้บอร์ด 1
3. **บอร์ด 3: Monitor & Control** – แสดงความคืบหน้าและจำนวนยาบนจอ 7‑segment, ควบคุมไฟ LED 4 สี, ควบคุมวาล์วและระบบล้าง เมื่อครบเป้าหมายจะส่ง RESET กลับไปบอร์ด 1

การสื่อสารระหว่างบอร์ดใช้รูปแบบข้อความง่าย ๆ:

| ทิศทาง | รูปแบบข้อความ | ความหมาย |
| --- | --- | --- |
| บอร์ด 2 → บอร์ด 1 | TARGET:xxx\n | กำหนดจำนวนเม็ดยาเป้าหมาย xxx (0–999) |
| บอร์ด 1 → บอร์ด 3 | PILLS:xxx\n | จำนวนเม็ดยาที่นับได้ปัจจุบัน |
| บอร์ด 1 → บอร์ด 3 | PROGRESS:yyy\n | ความคืบหน้าเป็นเปอร์เซ็นต์ (0–100) |
| บอร์ด 3 → บอร์ด 1 | RESET\n | แจ้งให้เริ่มรอบผลิตใหม่หลังการล้าง |

ทุกข้อความต้องจบด้วย \n และใช้ Even parity ตามโจทย์กำหนด

## โครงสร้างฮาร์ดแวร์และหน้าที่ของแต่ละบอร์ด

### บอร์ด 1 – Sensor & Display

**บล็อกที่ 1: Pill Counter & Production Display (20 คะแนน)**

* ใช้ **HC‑SR04** (พอร์ต PD2/TRIG และ PD3/ECHO) วัดระยะทางของเม็ดยาที่ไหลผ่านสายพาน – อ่านระยะ **สามครั้งติดกัน** แล้วตรวจว่าระยะ < 10 cm จึงถือว่ามีเม็ดผ่านหนึ่งเม็ด
* หลังตรวจพบเม็ด จะเพิ่มตัวนับ pill\_count (0–999) และล็อกการตรวจซ้ำด้วย **lockout 150 ms** เพื่อกันนับซ้ำจากเม็ดเดียวกัน
* แสดงผลบน **LCD 16×2** (RS → PB1, EN → PB2, D4‑D7 → PC4–PC7) ดังนี้:
* บรรทัด 1: Pills: xxx Target: yyy
* บรรทัด 2: Progress: zzz%
* คำนวณ progress\_percent = (pill\_count × 100) ÷ target\_count (ถ้ามีการตั้งเป้าหมาย) กำหนดช่วง 0–100%
* ใช้ **Timer1** ในโหมด CTC สร้าง tick 1 ms เพื่อใช้ใน lockout และ timeout
* มี **timeout 30 s** หากไม่ได้รับ/ส่ง serial ใด ๆ จะขึ้น “Timeout!” บน LCD และให้ LED13 กระพริบ

**บล็อกที่ 2: Communication Hub (15 คะแนน)**

* ใช้ **USART0** ที่ 9600 bps, 8E1 เชื่อมต่อกับบอร์ดอื่น
* รับคำสั่ง TARGET:xxx จากบอร์ด 2 → ตั้งเป้าหมาย target\_count
* ส่งสถานะไปบอร์ด 3 ทุก 200 ms เป็นสองบรรทัด: PILLS:xxx และ PROGRESS:yyy
* รับ RESET จากบอร์ด 3 → รีเซ็ตตัวนับและเริ่มรอบใหม่

โค้ดเต็มอยู่ในไฟล์ **board1\_sensor.c** ซึ่งมีการกำหนดพอร์ตและ ISR ตลอดจนฟังก์ชันช่วยต่าง ๆ เช่น measureCm(), lcd\_init(), usart0\_send\_string() เป็นต้น

### บอร์ด 2 – Input Control

**บล็อกที่ 3: Quality Control Input (20 คะแนน)**

* ใช้ **Keypad ADC 4×4** ต่อที่ช่อง A5 (PC5) พร้อมแรงดัน R‑ladder ตามโจทย์ ค่าแต่ละปุ่มอยู่ในช่วงดังตาราง (1 → 50–80, 2 → 90–120, 3 → 130–160, 4 → 170–200, 5 → 210–240, 6 → 250–280, 7 → 290–320, 8 → 330–360, 9 → 370–400, 0 → 410–440, \* → 450–480, # → 490–520)
* อ่านค่าด้วย ADC แล้วแม็พเป็นปุ่ม (0–9, \*, #)
* มี **debounce 50 ms** – อ่านค่า 2 ครั้งและทิ้งหากต่างกันมากกว่า 10 ขั้น ADC
* กดตัวเลข 0–9 เพื่อสะสมเป็นตัวเลขเป้าหมาย (0–999)
* กด \* เพื่อล้างค่า, กด # เพื่อตกลงและส่ง TARGET:xxx ไปบอร์ด 1
* มี **timeout 30 s** หากไม่มีการกดใด ๆ จะกลับสู่สถานะพร้อมรับคำสั่ง

**บล็อกที่ 4: User Interface Display (15 คะแนน)**

* ใช้ **LCD 16×2** ตัวที่ 2 (RS → PB1, EN → PB2, D4‑D7 → PC4–PC7) แสดงสถานะการป้อนเป้าหมาย
* แสดง Set Target: xxx บรรทัด 1 และ Press # to confirm บรรทัด 2
* ใช้ LED 3 ดวง (Green = PD4, Yellow = PD5, Red = PD6) แสดงสถานะ:
* **เขียว**: พร้อมรับคำสั่ง
* **เหลือง**: กำลังตั้งค่า
* **แดง**: ข้อผิดพลาด (เช่น ค่าเกินขอบเขต)
* ส่งข้อความผ่าน USART0 ไปบอร์ด 1 เฉพาะเมื่อกด #

โค้ดเต็มอยู่ในไฟล์ **board2\_input.c** ซึ่งประกอบด้วยการตั้งค่า ADC, debounce keypad, LCD display, LED status และการส่งข้อความ

### บอร์ด 3 – Monitor & Control

**บล็อกที่ 5: Production Monitor (15 คะแนน)**

* รับข้อความ PILLS:xxx และ PROGRESS:yyy จากบอร์ด 1 ผ่าน USART0 (9600 bps 8E1)
* แสดงผลบน **7‑segment display 4 หลัก** (แบบมัลติเพล็กซ์) โดยใช้ Timer2 interrupt ที่ 1 kHz
* **สองหลักซ้าย** – จำนวนเม็ดยา (0–99) ซึ่งเป็น pill\_count mod 100
* **สองหลักขวา** – เปอร์เซ็นต์ความคืบหน้า (0–99) ซึ่งเป็น progress\_percent mod 100
* ใช้ LED 4 ดวง (PB2–PB5) แสดงสถานะ:
* **เขียว**: ความคืบหน้า < 50%
* **เหลือง**: 50–89%
* **แดง**: 90–99%
* **น้ำเงิน**: กระพริบที่ ≈ 4 Hz เมื่อครบ 100%

**บล็อกที่ 6: Cleaning & Batch Control (15 คะแนน)**

* เมื่อ progress\_percent ≥ 100 จะเข้าสู่โหมดล้าง:
* เปิด **Packaging LED** ที่ PD2 นาน 5 s – จำลองการบรรจุขวด
* เปิด **Solenoid Valve** ที่ PD3 นาน 3 s – จำลองการล้างสายพาน
* แสดงคำว่า CLEn บน 7‑segment
* ส่ง RESET\n กลับไปบอร์ด 1
* เพิ่มตัวนับรอบการผลิต และเมื่อครบเวลาจะกลับสู่โหมด Run
* มี **timeout 30 s** ถ้าไม่ได้รับข้อมูลจากบอร์ด 1 จะเข้าสู่โหมด timeout: แสดง ---- และเปิดไฟแดงค้าง

โค้ดเต็มอยู่ในไฟล์ **board3\_monitor.c** มีการใช้งาน Timer0/Timer2 interrupt, multiplex 7‑segment, การกระพริบ LED น้ำเงิน และการควบคุมโหมดล้าง

## วิธีการพัฒนาระบบ

1. **วางแผนต่อวงจร** – จัดอุปกรณ์ตามผังในภาพ (HC‑SR04 ต่อกับบอร์ด 1, Keypad ต่อกับบอร์ด 2, 7‑segment และ solenoid ต่อกับบอร์ด 3) และใช้สายข้าม TX/RX เพื่อสื่อสารระหว่างบอร์ด พร้อม GND ร่วมกัน
2. **เขียนโค้ด** – แยกเป็นไฟล์ board1\_sensor.c, board2\_input.c, และ board3\_monitor.c ตามหน้าที่ พร้อมคอมเมนต์และฟังก์ชันช่วยเหลือ ได้แก่ Timer, ADC, LCD, Serial, Keypad, 7‑segment, LED และ Solenoid
3. **คอมไพล์และอัปโหลด** – ใช้ avr-gcc หรือ Arduino CLI ตั้งบอดเรตและพอร์ตให้ถูกต้อง เช่น -DF\_CPU=16000000UL -mmcu=atmega328p แล้วอัปโหลดไปแต่ละบอร์ด
4. **ทดสอบ** – เปิดแต่ละบอร์ดแยกกันเพื่อตรวจสอบว่าใช้งานได้ตรงตามบล็อก เช่น บอร์ด 1 นับเม็ดได้และแสดงข้อมูล, บอร์ด 2 รับค่าจากผู้ใช้และส่ง TARGET, บอร์ด 3 แสดงผลถูกต้องและควบคุมการล้าง
5. **เชื่อมต่อครบระบบ** – ทดสอบการสื่อสารทั้งสามบอร์ด: ตั้งเป้าผ่านบอร์ด 2 → บอร์ด 1 นับเม็ด/ส่งความคืบหน้า → บอร์ด 3 แสดงผล/ล้าง และรีเซ็ตกลับไปบอร์ด 1

## เอกสารและไฟล์ที่แนบมา

| ไฟล์ | รายละเอียด |
| --- | --- |
| board1\_sensor.c | โค้ดสำหรับบอร์ด 1: นับเม็ดยา, แสดงบน LCD และสื่อสารกับบอร์ดอื่น |
| board2\_input.c | โค้ดสำหรับบอร์ด 2: รับค่าจาก Keypad ADC, แสดงบน LCD และส่งเป้าหมายไปบอร์ด 1 |
| board3\_monitor.c | โค้ดสำหรับบอร์ด 3: แสดงข้อมูลบน 7‑segment, ควบคุม LED 4 สี, ควบคุมการบรรจุและล้าง |
| ภาพเชื่อมต่อ | รูป PNG แสดงการเชื่อมต่อระหว่างบอร์ดทั้งสามและอุปกรณ์ประกอบ |

โปรเจ็กต์นี้แสดงให้เห็นถึงการประยุกต์ใช้พื้นฐานการเขียนโปรแกรมไมโครคอนโทรลเลอร์ AVR (C register level) ตลอดจนการใช้งาน ADC, Timer/Interrupt, Serial communication, Keypad interface, LCD display, 7‑segment display, และการควบคุมอุปกรณ์ภายนอก เพื่อสร้างระบบควบคุมสายการผลิตยาแบบครบวงจร