

#	Sensor / Модел	Protocol Family (семејство)	MCU Primitive (какво прави MCU)	Needs External Lib?	RAW Output (от MCU)	Препоръчан начин на свързване към системата	Подходящи контролери (препоръка)	Бележки
1	A02YYUW Waterproof Ultrasonic (SEN0311)	UART	UART_READ_DISTANCE – чете кадри @ 9600 bps и вади дистанция/статус	Да (лека UART/парсер логика, но не тежка библиотека)	Число (разстояние) + евентуално статус код	MCU ↔ RS232/UART → USB/Serial към хоста или през ESP32/WiFi	ESP32 / FireBeetle 2 ESP32-E, Arduino Mega; (Uno/Nano – ОК, но само ако UART не е споделен с хоста)	По-добре на платки с повече UART-и (ESP32/Mega), за да няма SoftwareSerial хаос.
2	Capacitive Soil Moisture (SEN0193)	ANALOG	READ_ANALOG – просто измерване на напрежение (0–3V)	Не	Цяло/float ADC стойност	MCU → USB Serial или WiFi; няма специални изисквания	Arduino Uno/Nano/Mega, Wemos/NodeMCU, ESP32, FireBeetle 2 ESP32-E	Цялата „умност“ (влага %, калибрация) е в backend.
3	Gravity Water Flow G1/2 (SEN0550)	PULSE / FLOW	MEASURE_PULSE_RATE – брой NPN импулси за прозорец от време	Не	Pulse count / pulses per window	MCU → USB Serial / WiFi; самият сензор към отделен дигитален пин	Arduino Mega, ESP32, FireBeetle 2 ESP32-E; Uno/Nano – ОК за 1–2 flow сензора	Backend превръща импулси → L/h (формула 1L=150 импулса).
4	RS485 PAR Sensor (SEN0641)	RS485 / Modbus-RTU	MODBUS_RTU_READ_REGS – изпраща Modbus заявка и връща регистри	Да (Modbus RTU стек)	Масив регистри / сурови байтове	MCU ↔ RS485 адаптер → хост; силно препоръчително отделен хардуерен UART	Arduino Mega, ESP32, FireBeetle 2 ESP32-E; избягваме Uno/Nano и Uno R4 WiFi за продукционен Modbus	В DeviceTemplate отбелязваме, че изисква контролер с „dedicated UART + RS485 module“.
5	Analog EC Sensor V2 (DFR0300)	ANALOG	READ_ANALOG – измерва 0–3.4V изход	Не	ADC стойност	Стандартно MCU → USB/WiFi	Uno/Nano/Mega, Wemos/NodeMCU, ESP32, FireBeetle 2 ESP32-E	EC формулата, температурна компенсация и калибрация са изцяло в backend.
6	Analog pH Sensor V2 (SEN0161-V2)	ANALOG	READ_ANALOG – измерва 0–3.0V изход	Не	ADC стойност	Стандартно MCU → USB/WiFi	Uno/Nano/Mega, Wemos/NodeMCU, ESP32, FireBeetle 2 ESP32-E	pH(...) формулата и двучоквата калибрация са в backend (може да ползваме идеите от официалната софтуерна библиотека, но от страната на сървъра).
7	DHT22 Temp & Humidity (SEN0137)	DHT / Single-Wire Pulse	DHT_READ – time-critical pulse протокол по един пин	Да (малка DHT библиотека)	Темп + влажност (float)	MCU → USB/WiFi; DHT към отделен дигитален пин	Uno/Nano/Mega, ESP32, FireBeetle 2 ESP32-E; Wemos/NodeMCU – става, но с внимание към тайминги	Това е отделно „семејство“ от OneWire – в firmware пазим отделен модул (както във v4 SINGLE_WIRE_PULSE).
8	DS18B20 Waterproof (DFR0198)	ONE_WIRE	ONEWIRE_READ_TEMP – 1-Wire протокол, адрес + четене	Да (OneWire lib, малка)	Температура (float)	MCU → USB/WiFi; няколко DS18B20 могат да споделят пин	Uno/Nano/Mega, ESP32, FireBeetle 2 ESP32-E; Wemos/NodeMCU – позволено	Втори отделен модул във firmware (SINGLE_WIRE_ONEWIRE), различен от DHT.
9	URM09 Ultrasonic (Trig, SEN0388)	TRIG-ECHO / PULSE	ULTRASONIC_TRIG_ECHO – дава trigger, мери high-pulse време на ехо	Не	Pulse duration → конвертира се до разстояние	MCU → USB/WiFi; ползва 1 дигитален пин (вход/изход multiplex)	Uno/Nano/Mega, ESP32, FireBeetle 2 ESP32-E	Подобен на HC-SR04 логика; може да споделя общ „ultrasonic“ примитив с други.
9.1	URM09 Ultrasonic (Analog, SEN0307)	ANALOG	READ_ANALOG – аналогов напрежение → дистанция	Не	ADC стойност	Стандартно MCU → USB/WiFi	Uno/Nano/Mega, Wemos/NodeMCU, ESP32, FireBeetle 2 ESP32-E	Формулата Distance = Vout*520/Vin е в backend, не във firmware.