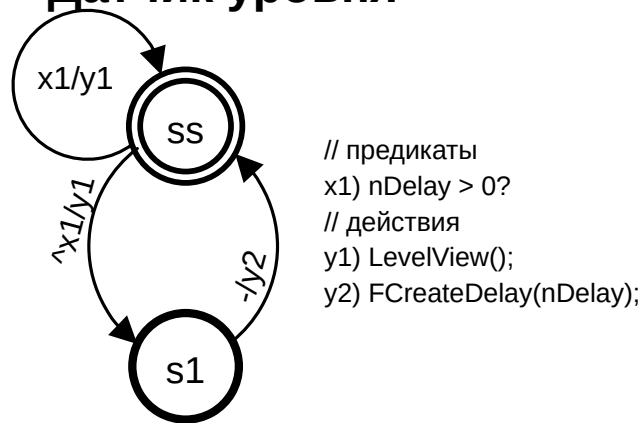
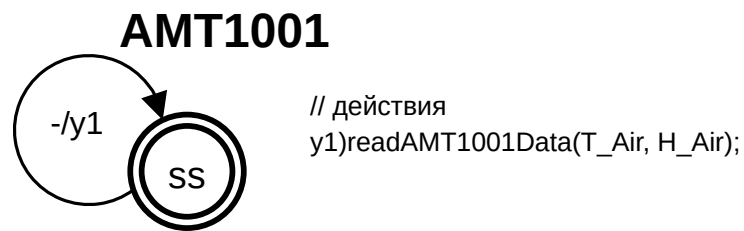


## Датчик уровня



### Краткое описание алгоритма

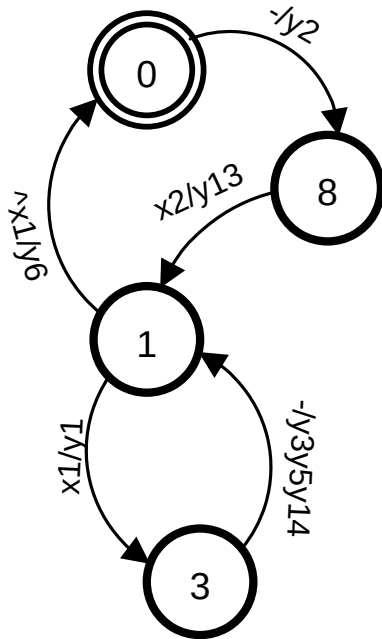
В начальном состоянии SS предикат x1 анализирует величину задержки. При истинном значении, когда она равно нулю, в цикле выполняется действие y1, которое анализирует входы датчика и отображает их состояния. В рамках действия y1 выполняется чтение входов датчика и выводится в порт состояние входов, изменивших свое значение, что и реализует функция LevelView(). Действие y2 запускает заданную задержку между опросами датчика. Это реализует библиотечная функция ядра FCreateDelay(nDelay), где nDelay- величина задержки.



## Краткое описание алгоритма

В цикле происходит вызов метода readAMT1001Data(T\_Air, H\_Air), который возвращает температуру - T\_Air и влажность - H\_Air. Последние являются свойствами класса AMT1001. Подробнее смотреть – заголовок и реализацию класса.

## EC\_meter



x1)  $i < 10; \text{var} < \text{nCyclesEC} ?$   
 x2)  $(\text{millis}() - \text{lastEcRead}) \geq \text{nTimeEC} ?$   
 x5)  $\text{PHAndTemperature.FGetState}() == "0" ?$

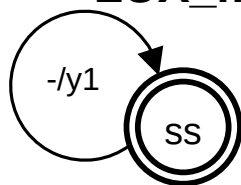
y1) `digitalWrite(d2, HIGH);`  
       `digitalWrite(d1, LOW);`  
 y2) `lastEcRead = millis();`  
 y3) `dDap = analogRead(a1);`  
       `dDATap = dDap;`  
       `ap = 0 + dDATap + ap;`  
       `digitalWrite(d1, HIGH);`  
       `digitalWrite(d2, LOW);`  
 y5) `dDan = analogRead(a1);`  
       `dDATan = dDan;`  
       `an = (0 + dDATan) + an;`  
       `digitalWrite(d1, LOW);`  
       `var++;`  
 y6) `digitalWrite(d1, LOW);`  
       `digitalWrite(d2, LOW);`  
       `float ap_raw=(float)ap/var;`  
       `float an_raw=(float)an/var;`  
 ...  
 y13) `pinMode(d1, OUTPUT);`  
       `pinMode(d2, OUTPUT);`  
       `var = 0; ap = 0, an = 0;`  
 y14) `pinMode(d1, OUTPUT);`  
       `pinMode(d2, OUTPUT);`

### Краткое описание алгоритма

Алгоритм начинает работу, когда завершил свою работу алгоритм измерения PH (x5) (измерения PH и температуры). Затем выполняется заданное число циклов измерения ЕС. Каждый цикл измерения включает два измерения электропроводности в прямом и обратном направлении. В заключении в рамках действия y6 Выполняется истинное значение электропроводности.

**Замечание.** В рамках начального запуска (см. `setup()`) запускается метод `setup_Ec_meter`, реализующий функции начальной установки для измерения ЕС.

## LUX\_meter



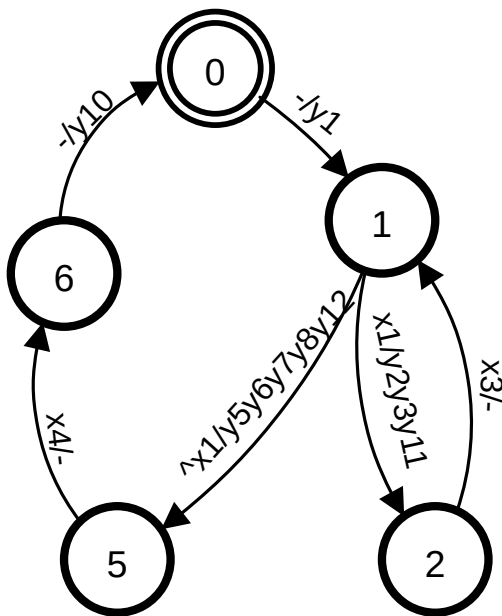
// действия

y1) Измерение освещенности;

### Краткое описание алгоритма

В цикле в рамках метода y1(). Здесь выполняются замеры освещенности в зависимости от заданного режима измерения. Сам режим измерения задается извне и определяется методом LUXMeterControl(), который циклически запускается в рамках работы функции loop() головного модуля проекта.

## PhAndTemperature



x1) i<10;?

x2) (i<8);?

x3) (millis() - lastTime30) >= 30?

x4) (millis() - lastTime30) >= 500?

x5) EC\_Meter.FGetState() == "0"?

y1) i=0;

y2) i++;

y3) buf[i] = analogRead(analogInPin);

y5) ;

y6) avgValue = 0;

y7) for (int i = 2; i < 8; i++) avgValue += buf[i];

y8) Расчет PH\_acc;

y10) readTemperaturePH();

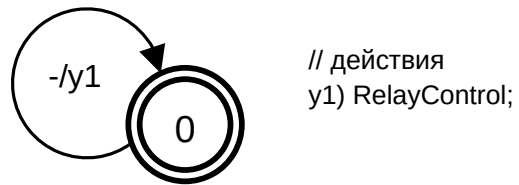
y11) lastTime30 = millis();

y12) lastTime500 = millis();

### Краткое описание алгоритма

Алгоритм начинает работу, когда завершил свою работу алгоритм измерения ЕС (измерения электропроводности). Затем выполняется заданное число циклов измерения PH. Измеренные значения помещаются в массив измерений. Далее последовательно выполняется ряд действий: массив сортируется - действие y5; Отбрасывается минимальное и максимальное значения и оставшиеся суммируются – y7; значения усредняются приводятся к заданной форме – y8. В заключение реализуется задержка между измерениями – переход 5->6->0.

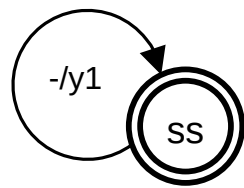
# RelayControl



## Краткое описание алгоритма

Управление реле сводится к циклическому запуску функции RelayControl().? Которая сравнивает текущее состояние реле со значением полученным извне. Далее состояние реле приводится в соответствии с заданным значением.

## SensorDS18B20



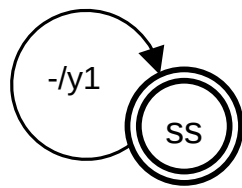
// действия

y1) Измерение температуры;

### Краткое описание алгоритма

В цикле выполняются замеры температуры, реализуемые в рамках внешней библиотеки работы с датчиком DS18B20.

## SensorLevel



// действия  
у1)Измерение значения  
уровня жидкости;

### Краткое описание алгоритма

В рамках действия у1 происходит опрос трех цифровых входов отражающий состояния текущего уровня жидкости. Их текущая комбинация сосостояния преобразуется в уровень жидкости, выраженный процентами наполнения бака.