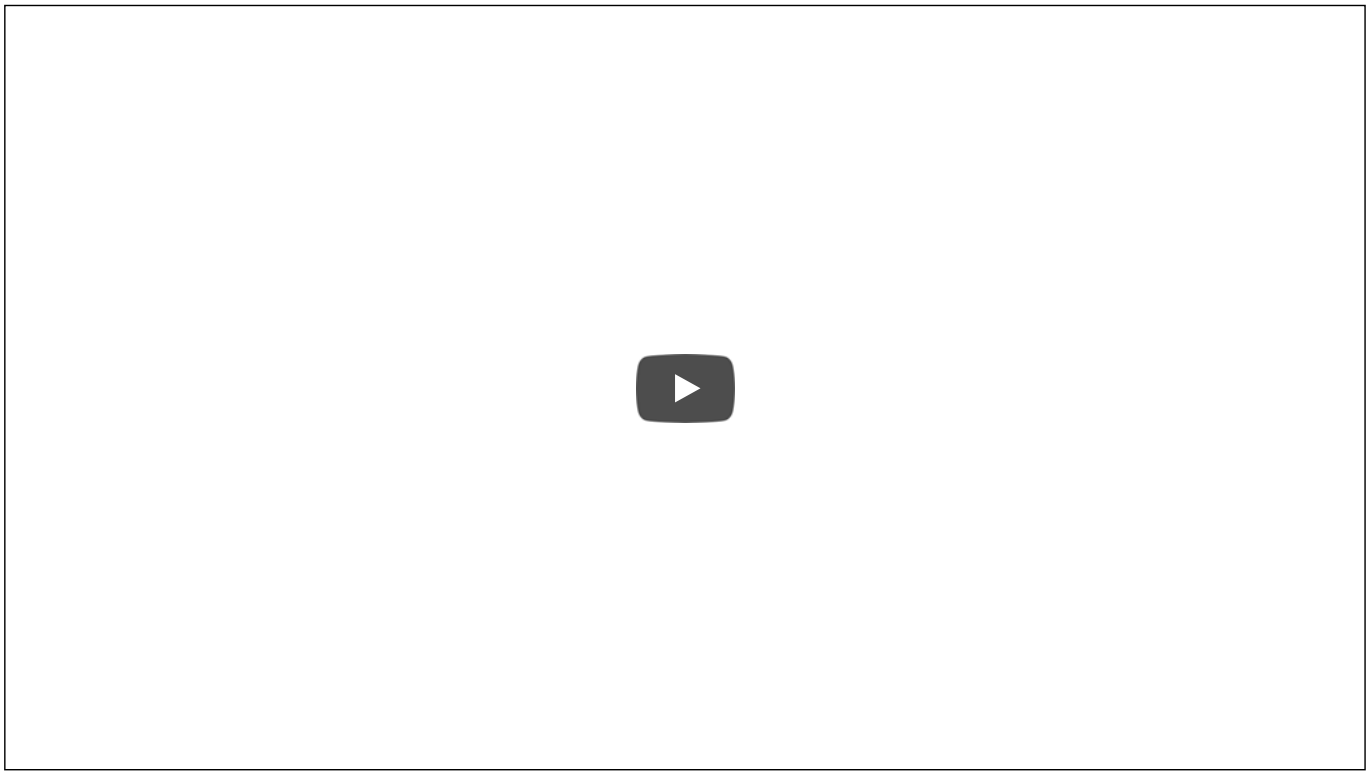
Unique Crossing

This really working design was developed a two year ago, and reliably works to this day. The actuators of the barriers is independent, since each barrier is driven by a separate servo. Since barriers can be moved asynchronously and at different speeds, it's more like the real world in which it happens that way. Also in the module use classical flashing lights.



All crossing rail electronics made on the one autonomous URB unit with the possibility of connecting to the URB bus. The mechanism is made in such a way as to fit completely into the space under the crossing of the move and not to leave its dimensions.

To automatically lower the barriers when passing a train, the original algorithm given below and two conventional infrared sensors Arduino are used. The design of these sensors is modified and described in the Sensors chapter.

Electronics

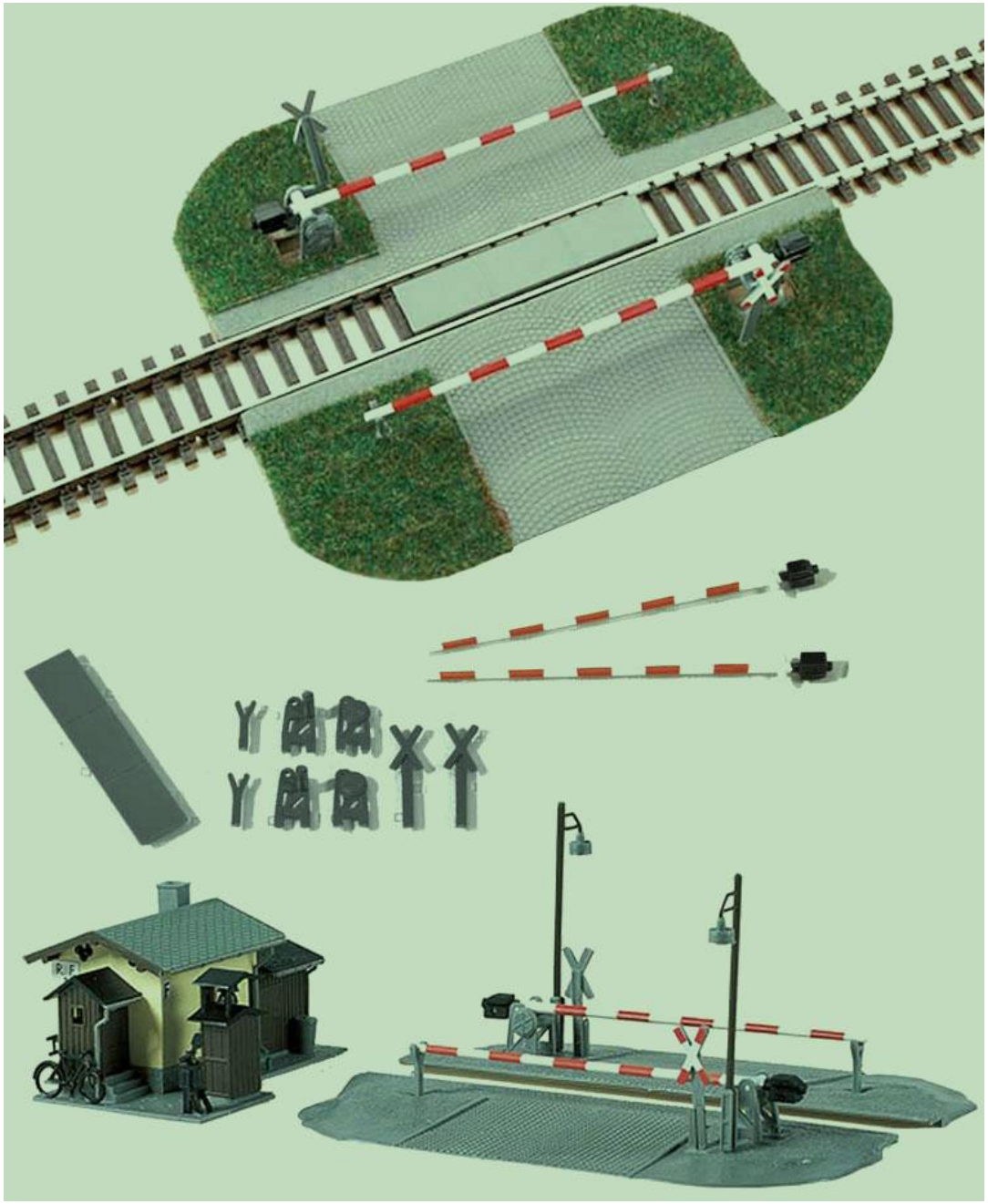
The sources of the train signal about the approach and the departure from the wide railway crossing area will be infrared sensors located at a distance from the crossing. I suggest that you will establish the distance between them experimentally.

The normal servo always rotates at its maximum speed. For different models of servo drives speed is different, there are fast and slow servos, but the speed of rotation is always constant. That is, we can not make the rotation faster than the installed by the servo manufacturer with the help of electronic control (though we can increase the speed mechanically using the reduction gear). But we can lower this speed by rotating the servo shaft to a small angle with a delay, and repeat this process again and again until the servo turns to the desired position.

For this I propose to use the alternate library for Arduino VarSpeedServo, very similar in function to the standard Servo library. You will read instruction how install the third-party library in the Arduino IDE environment on the author's page. As a result, you get another setting for controlling the rotation of the servo. Changing the command myservo.write (angle, speed, true); the second parameter you can set the speed from very slow: 10-50, to maximum: 255.

Just in case, I will explain, under all the processes when using Arduino it is easier to manage with the help of variables. And servo control is no exception. In the sketch for the motor-point for the junction, it is more convenient to specify directly the angle of rotation of the servo, but in other cases it is better to use the type variables INT (integer). Variables

can be calculated mathematically when the program is running. Among other things, this allows create realistic behavior of moving objects on the layout, for example, when the position of the descending barrier is closer to stopping at the lowest point, it can be slowed down or a small rebound can be realized when the barrier reaches the extreme positions.



It's design based of the simplest Auhagen 41582 Level crossing kit. Since there are many similar sets, you may use any. And of course, this module work both in automatic mode and in manual controlled mode.

The DIY super slim lifting gate mechanism for railway crossing

The mechanism is made in such a way as to fit completely into the space under the crossing of the move and not to leave its dimensions. This mechanism converts the translational motion of the thrust from the servo to the rotation of the barrier and at the same time has a damper. The horizontal angle of the thrust vector can be set by you in a wide range.

This level crossing module has advanced settings: you can raise barriers synchronously, or with time-sharing – for example the barrier starts to go down (or go up) on one side of the crossing, and on the other side the same thing happens, but with a half-second delay. Also you independently regulate the speed of lifting or lowering barriers on both sides of the crossing. All light effects are regulated – the frequency of flashing of the signals, the sequence and delay of the light signaling. You can for even joke to raise one barrier and lower the other.

Crossing Sketch

Algorithm and sketch of railway crossing

I have already described the use of Arduino sensors for use in railway electronics and problems with their interaction with trains. To overcome these problems, several methods should be used at once: physicals and algorithmics.

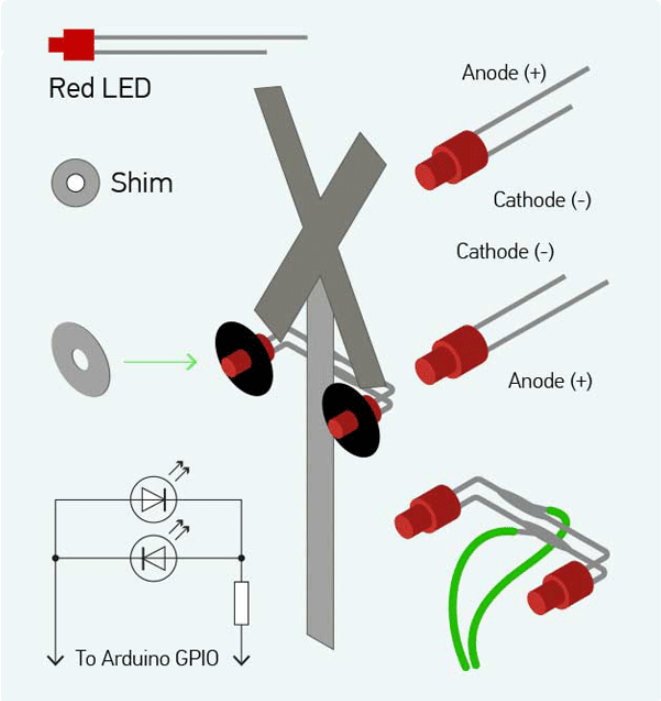
Here I use infrared sensors my module with a separated emitter and receiver deployed relative to the rails by 45 degrees and spaced apart in height. Thus, I remove problems with different reflective ability of painting cars and false sensor triggering when changing moving cars of a train in the sensor operating zone. That is, there is an inverse algorithm, unlike the usual one – while the train does not exist we have a sensor active, when the train crosses the sensor, and until it completely leaves its area of operation, the sensor is blocked. Therefore, the setting of the variable resistor on IR sensor housing should be set to the minimum position for its activation.

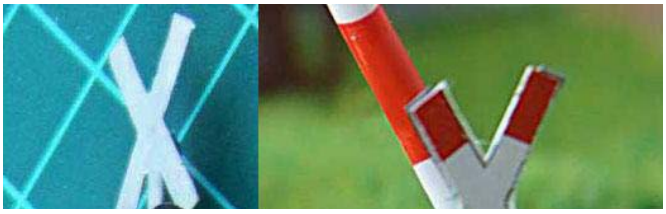
It is obvious that the minimum Trigger distance from the sensor zone before crossing point should be such that when the train approaches the maximum speed the barrier had time to descend. This distance should be established experimentally. But Arduino gives us a lot of extra features, for example, after triggering the sensor, we can turn on the blinking alarm light at crossing , and with a certain delay after that we lower the barrier, in this case the length of the Trigger distance needs to be increased.

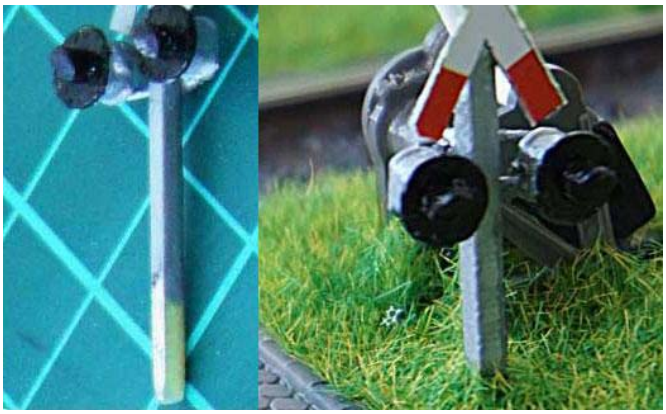
The original algorithm for automate moving barriers uses two variables for each sensor, one of which acts as a latch. As a result, the time independence is ensured and the code becomes very simple.

```
if (latch_s1 && latch_s2 && !trigger_s1 && !trigger_s2) {
latch_s1 = false;
latch_s2 = false;
}
```

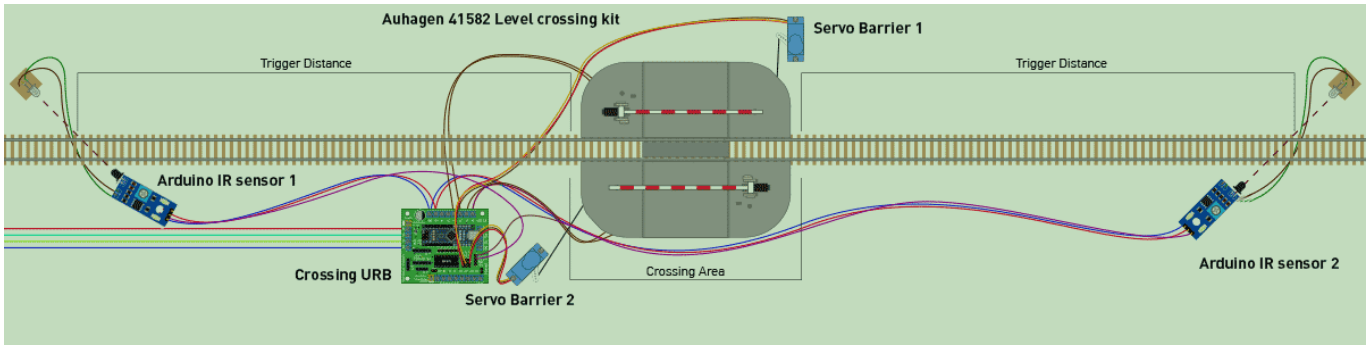
The algorithm correctly reacts to the stop of the train in the sensor area and even the train maneuvers at the rail crossing.







Blinking signals are collected on red LEDs of a two-cylindrical shape. Anodes and cathodes of these LEDs are cross-connected. As a result, with the opposite change in the state of the two outputs of Arduino from high to low levels, only one of the two LEDs will light.



To configure the module, I made a separate sketch. By sending protocol commands 2 through the computer's serial terminal, you can adjust the angles of the barriers, their behavior and speed.

Adjustments Crossing Sketch

Relay Cabinet