

```

void GeneralControlLoop::init()
{

    localParameters.outputNegMax=-4000; //-4000 is -100%
    localParameters.outputPosMax=4000; // 4000 is +100%
    localParameters.polyPar[0]=0; //adjust those 4 values to your spring
    localParameters.polyPar[1]=1; //should map displacement into a force in [N]
    localParameters.polyPar[2]=0; // here a linear mapping with a gain of 1
    localParameters.polyPar[3]=0;

    localParameters.radPerEncoderCount=2*M_PI/(2000.0*53.0); //gear box ratio of 53 and
        //2000 counts per rotation with encoder (4 x 500 pulse, edge count)

    localParameters.spNegMax=-10; //limit position reference value
    localParameters.spPosMax=10;
    localParameters.tag=0; //don't care, leave to 0
    localParameters.timePeriod=1010; //us
    localParameters.torqueConstant=1; // motor constant, set to useful value when torque control required

    localParameters.params.pidParameters.IntegralNegMax=-4000; //limit the integrator
    localParameters.params.pidParameters.IntegralPosMax=4000;
    localParameters.params.pidParameters.deadBand=0; //choose dead-band
    localParameters.params.pidParameters.forwardGain=0; //set controller gains
    localParameters.params.pidParameters.pgain=0.12;
    localParameters.params.pidParameters.igain=0;
    localParameters.params.pidParameters.dgain=0;
    localParameters.params.pidParameters.integral=0; //initialise the integrator
    localParameters.params.pidParameters.lastError=0; //initialise last error

    //configure a position controller
    p_robot->getGanglion(0)->getMuscles()[0]->setControllerParams(Position,localParameters);
}

void GeneralControlLoop::cycle() //cyclic loop
{
    if (! p_robot->controlparameterRequestQueueEmpty())
    {
        //we should not run the controllers if there are still parameter set
        // requests in the queue
        std::cout<<"Serving Queue."<<endl;
    }
    else
    {
        double referencePosition=0;

        //do some useful control here

        p_robot->getGanglion(0)->getMuscles()[0]->enableController();
        p_robot->getGanglion(0)->getMuscles()[0]->setControllerRef(Position, referencePosition);
    }
}

```