Shift4Me build manual

Introduction

So, you're planning on building a Shift4Me? Great! It's fun to do and even greater to ride your bike once it's installed. Before you go any further, please read through this little introduction to avoid disappointments.

First of all: this manual can be used if you want to build a Shift4Me as you see in the demo video's on the website. If you want to build one that is different(*) just get the "PCB" file and start from there. Or maybe you want another design of the BigBox? Then just disregard the 3Dprint section and mount the system in your own fashion.

Building a Shift4Me is pretty straight-forward if you have (intermediate) soldering skills, you are (a little bit) familiar with Arduino programming and you have (some) experience on bicycle mechanics. Oh, you don't? Then there's two options: 1-just go ahead but be prepared for some trial-and-error. 2-pick up the phone and contact someone who has the experience. Don't be shy - I'm sure all handy people you know will be glad to help out with this fun project.

What you're getting yourself into is:

- Print the 3D parts
- Shop for parts
- Solder everything
- Test connections
- Install everything on your bike
- Surprise your neighbours with your self-build auto-gearing bicycle

Building a Shift4Me is going to cost you about 120 euro/USD in parts (excluding the bike itself of course), and is going to take you about two days to finish. The reward will be an automatic-gearing bicycle that is fun to ride.
A complete step-by-step build manual is yet to be finished. This one just has the basics covered.
Don't forget to post your experiences on the forum of the Shift4me website!

Have fun!

Jan Oelbrandt
Shift4Me developper

* People often ask me "Why don't you add a LCD screen, to display the current gear, battery status, and so on?" and things like "Add a speedometer, that would be a big improvement." I don't disagree with them, but I do have my reasons for not doing this (see the "Shift4Me introduction"). But if you want to add bells and whistles to Shift4me: who's gonna stop you? Just go ahead!

Content

1. Prepare your bike

    Make sure you have a good working bicycle ready. That means: aside from basics such as good brakes, proper tires etc. it should have a smoothly shifting and correct handling gearing system. Any bike with a standard derailleur (any amount of sprockets) or a hub gear (any amount of gears) will do, as long as it's in working order. Only gearing systems using a single-cable control, spring-return can be used (that means: no Rohloff equipped bike, sorry folks!).
    Don't dismantle anything yet on the bike, let's build the Shift4Me first.

2. Get your stuff together
    a. Tools
        i. Soldering iron & lead to… solder
        ii. Pliers to mount the plugs
        iii. A clean desk
        iv. A multi-meter. Preferably one that has a beep-function for testing zero ohm (contact test)
        v. A computer with the Arduino software installed (the IDE). Feel free to choose the desktop or the online version. https://www.arduino.cc/
        Preferably a laptop, since it's going to be necessary to have it nearby your bike during setup. Remember: when using the local version, all script files have to be present in the right folder on your computer. See www.arduino.cc for all documentation.
    b. Components

        You need all the items to build the Shift4Me described in the "basic components" pdf file. The `BigBox` is designed to accommodate various connectors. If you want to use other types of connectors, redraw the `BigBox's` STL file first.
    c. PCB layout, routing info etc… It's all in the PDF files called "PCB" and "schematic" in the Downloads on the Shift4Me website.
    d. Get the STL files to be able to 3D-print the BigBox, the control unit and the magnet holder. They're in the "3D-prints" zip file in the Downloads.
    e. Get the right STL file for the servo horn to match your gearing system (Hubgear or Derailleur)
    f. Download the Arduino scripts from the "Scripts" zip file (again: choose Hubgear or Derailleur)

3. Prepare the Arduino

   a. Shift4Me is designed around an Arduino Nano. If you want to use an other board, you're on your own to redesign the PCB. The required specs, if you choose an other board, can be found in the Appendix below.
   b. Unpack the `Arduino` Nano board and connect it to your computer. The power-LED should light up on the board. Now startup the Arduino IDE and upload the test script called "Blink". You don't have to download it seperately, it is under the "File->Examples->Basics" menu in the IDE. Once uploaded, an other LED on the board should go on and off repeatedly.
      i. Does it work? Is the LED of your board blinking as it should? OK then, proceed to step 3.
      ii. No luck? No blinking LED? Error messages popping up in the Arduino IDE? Don't worry, there's a whole community of Arduino users ready to help you at www.arduino.cc (if you don't find what you need on the Shift4Me forum)
   c. Get your Script files from the website if you haven't already done so. There a 3 script you're going to use: "tester", "calibrate" and also the appropriate script for your gearing system ("hubgear" or "derailleur"). Don't forget: if working locally, these should be in the right folder on your computer.
4. 3D print the `BigBox`, the `ServoHorn, MagnetHolder` and the `ControlUnit`
   a. All parts that have to be printed are in STL format. They first have to be saved as GCODE files, and a slicing software has to make sure that all overhangs are taken care of. If you're new to 3D-printing, please let someone assist you. Even better: contact your local 3D-printservice, they'll take care of everything.
   b. Printing material: PLA is fine. You can print everything *solid*, but *hollow* fine-mesh is sufficient.
   c. When printing is done, remove all support material (if the 3D-printservice hasn't already done so). Make sure you free the rectangular holes in the bottom of the `BigBox`, two tie-wraps will have to be able to feed through them.
   d. Check if these components will fit in the holes:
      i. For the `ControlUnit` - the `LED/button` combo
      ii. For the `BigBox`: side wall: the `ChargerPlug` and the `KeySwitch`. Front wall: `cable adjuster, female mini-jack chassis & male DIN-chassis`.
5. Assemble and solder the main PCB
   content will be available in the yet-to-be-completed manual -
6. Solder plugs
   content will be available in the yet-to-be-completed manual -
7. Assemble and solder the control PCB
   - *content will be available in the yet-to-be-completed manual* -
8. Solder the TestSwitch
   - *content will be available in the yet-to-be-completed manual* -
9. Mount everything in the BigBox
   - *content will be available in the yet-to-be-completed manual* -

10. Run some tests
    a. Test for wanted and unwanted circuit lines. Perform these steps using a ohm-meter or cable tester. (“short”=contact=0ohm and “NC”=no contact=infinite ohm). Set the `KeySwitch` to the “off” position. If you’re not sure which position that is, use your multimeter to determine (Key#1-Key#2 should NC). Also, connect the `ControlUnit` to the `BigBox` with the DIN connector
    b. You should have a “short” between the following. GND is the looped wire on your main PCB.
        i. GND -> `mini jack` outer shell
        ii. GND -> `screw terminal` #2
        iii. GND -> point#21
        iv. GND -> `charge plug` outer shell (with the charger plugged in)
        v. GND -> `ServoPlug` (black or brown wire)
        vi. `screw terminal` #3 -> `charge chassis` tip
    c. You should have NC between the following:
        i. `screw terminal` #1 and any of the `Arduino` pins
        ii. `screw terminal` #3 and any of the `Arduino` pins
        iii. `Charge chassis` outer shell -> `charge chassis` tip
        iv. Between any of the 3 `screw terminals`
        v. `screw terminal` #1 -> `ServoPlug`, middle leg (red wire)
    d. Now turn the key to the “On” position. You should have a “short” between `screw terminal` #1 -> `ServoPlug`, middle leg (red wire).
    e. DO NOT PROCEED if any of the above failed. If something turned out wrong, check for sloppy soldering, broken wires, etcetera.
    f. If you want to do additional checking, you can measure all the points of the `Arduino` to where they should “short”.
    g. Almost done… you’re ready now to run the TestScript. First, connect the `Sensor` and keep the `magnet` ready. Also plug the `TestSwitch` in the mini-jack plug on the main PCB. Now connect the `Arduino` to your computer using an USB cable and upload the script called “Shift4Me_Tester”. Immediately after you’ve done that:
        i. The buzzer should sound in a sequence
        ii. The led should light up (flashing)
        iii. Now display the Serial Monitor in the IDE. Now you can monitor the activity of the Sensor (hold your magnet on and off nearby the Sensor) and the TestSwitch. Also you can push the button on your `ControlUnit`. Everything should read out accordingly in the Serial Monitor. The “battery status” should read no more than 4V. See pic 9 for an example.
    h. If everything reads out fine in the Serial Monitor and you heard the Buzzer, saw the LED flashing, you can now upload the “Calibrate” script. After that, test signals will stop and you can finish the mounting in the `BigBox`. Unplug the `TestSwitch`, the `Sensor` and `DIN plug`.

        i. Put some non-conductive damping material to secure the Main PCB. Use some foam or a piece of garbage cloth.
        ii. Solder 8cm of thick wire to your battery packs (red=positive, black=negative). Tape them up to prevent shortcutting to the components. BEWARE: don't shortcut a battery pack, that can lead to accidents. Strip one wire at a time!
        iii. Take the `6V battery pack` and connect its positive side to the negative side of the `2.4V pack`, screw them in `screw terminal` #1
        iv. Screw the red wire from the `2.4V pack` to `screw terminal` #3
        v. Take a fourth `screw terminal` (mark it… #4) and use it to screw the black of the `6V pack` to the only loose wire left (the thick black one).
    i. No fire? No explosion? Then proceed… Plug the USB back into the mainPCB and connect the `TestSwitch`. Now open the Serial Monitor in the Calibrate script. Put the `KeySwitch` in the "on" position. Now when you flip the `TestSwitch` to one side, you should see the "servo pulsewidth" value going up while the `servo` should rotate in small steps. And have you guessed? The value should go down (the `servo` rotating in the other direction) when you flip the other side of the `TestSwitch`. Battery value should be well over 7V (if not, you'll have to charge first). See screenshot as in pic 10. All going well? Then leave the system in the state of "you reached the minimum position of the servo".

11. Install the BigBox on your bike
    a. Find a suitable place to mount the `BigBox` to your bike. Keep in mind that, once it is mounted:
        i. You have to be able to connect to the USB port of the `Arduino`.
        ii. The routing of the derailleur cable (that will go from the front of the `BigBox` to your derailleur or hubgear) is neat. That means: as little bends as possible and not interfering with moving parts while riding.
    b. Take two tie-wraps, feed them through the holes in the bottom of the `BigBox` and secure the `BigBox` to a bar of your bike. I like to mount it on the upright bar and I use a piece of old inner tube that I put at the back to protect the paint. See pic 11
12. Connect your hub gear or derailleur to the BigBox
    a. If you have a bike stand to lift the complete bike, put it up there. If not, think of a way to hang up the bike to the (back) wheel can turn.
    b. If you have a hubgear, it's a good idea to manually shift down (full cable release), then disconnect the cable with the fixing screw still attached, and write down the distance between center of the fixing screw and the end of the hose. For example: with Shimano Nexus 8 this distance is 10cm. When mounting the new cable, you'll need this.
    c. Dismount the complete manual derailleur cable from your bike.
    d. Take a new derailleur cable that's long enough (duh!) to go from the `BigBox` to your derailleur or hubgear. The hose: long enough to go from the `BigBox` to a fixed point of your bike (see pics 12 and 13)

e. Installation of the cable to the `ServoHorn` is different because of the different design for derailleur vs. hubgear.
    i. Hubgear: Feed the cable through the `ServoHorn`, see pic 16. Feed it all the way so that the ball end is at the horn, see pic 14.
    ii. Derailleur: Feed the cable through the round spline that came along with the servo and mount it to the servo together with the `ServoHorn`, see pic 17.

f. Be careful to position the `ServoHorn` and the round (derailleur) or 6-tooth (hubgear) part so the cable is positioned as in pic 14. Also, the `servo` must be in “minimum position”. If you’re not sure, just turn the Shift4Me on (using your key) and it will set the `servo` at minimum position. Now feed the cable from the inside of the `BigBox` through the `cable adjuster` (sometimes it’s easier if you screw it off fist, then feed the cable through, and screw the adjuster back on)

g. Feed the cable through the hose (don’t forget the hose ends!). The hose should slide easily over the cable. Apply some grease.

h. Turn the Shift4Me on and plug in your `Testswitch`.

i. Make sure your the `cable adjuster` at the `BigBox` and the cable adjuster on your derailleur (you probably don’t see one if you have a hubgear) are screwed in all the way.

j. Pull the cable and screw/fix it to your derailleur or hubgear.

k. Now flip the `TestSwitch` so the `servo` begins to turn counterclockwise in small steps. Your derailleur or hubgear should start moving too as the cable is pulled by the `servo`. When you have a derailleur, your pedals need to be turning as the derailleur moves to a larger sprocket. With hubgears, this is not necessary.

13. Install the Control unit, the Sensor and the magnet
    a. Take two rubber bands and fix the `ControlUnit` to your handle bar. Lead the wire to the `BigBox` and connect the `DIN plug`.
    b. Screw the `magnet` to the `magnet holder`. Use two small tie-wraps to mount it to the inside of your left pedal.
    c. Solder the mono `mini-jack plug` to the `Sensor`, leave plenty of cable to reach the `BigBox`.
    d. Take one rubber band and mount the `Sensor` to the left horizontal bar. When you turn the pedals, the magnet should go past the `Sensor` with a distance of no more than 5mm. Plug the `Sensor` (the `mini-jack plug`, that is) in the `BigBox`.

14. Tune the system to your bike
    a. It’s a good idea, now that everything’s connected, to test the system once more. So take your laptop and put it close to the bike, connect via USB. Upload the TestScript you have used in section 10.g. Open the Serial Monitor, it should read “hall triggered” when you turn the pedals, you see the LED flashing and hear the buzzer. When flipping the `TestSwitch`, it should read "manual switch position: pull (or release) cable".

b. Now disconnect the `Sensor plug` from the `BigBox` and upload the "Calibrate" script. Leave the USB connected. Display the Serial Monitor.
c. Flip the `TestSwitch` until the derailleur cable it fully pulled (*). This is different for derailleur vs hubgear bikes:
   i. When fully pulled, a derailleur will be where the chain is in the largest sprocket (smallest gear).
   ii. When fully pulled, most hubgears will be in the highest gear.
d. At this point, when the cable is fully pulled, your Serial Monitor should be around 2300 up to 2400 pulsewidth. If this is a lot lower (say, 1800), your cable needs to be loosened. If the monitor displays "maximum reached" and your cable is not fully pulled, then you need to use one or both of the cable adjusters to tighten it up. *(*) fully pulled means: your derailleur is at the maximum of its travel.*
e. Success? Then let's check if your range is covered. Flip the `TestSwitch` over until the `servo` is in minimum position, your derailleur should be in the beginning of its travel. OK? Then flip the switch over until the `servo` is in maximum, and guess what? Your derailleur should be in maximum position. If this fails, you can try to fool around with the cable adjusters. If that still fails, you'll have to 3Dprint a `ServoHorn` of a bigger diameter to cover a wider range. In my test bikes, If have a 16mm diameter for a 9-speed Shimano sprocket, and a 28 diameter for a Shimano Nexus8 hubgear. This is because the travel (distance of cable movement) is different.
f. If you can cover the whole range of your gears, then you can move on to tune the gears. Again, this is different for derailleur (see 14.h) vs. hubgears (see 14.g):
g. For Shimano 7-8 speed hubgears it is very easy, because they have a fixed "spacing" and both calibrate in gear#4.
   i. Upload the script called "Hubgear" to the Arduino. When done, Turn the system on. The `servo` is now in "starting gear" which is also the calibrate position for the Shimano hubgear. There are two yellow line in the hub that should line up. If not, adjust the cable adjusters until they do.
   ii. The "Hubgear" script is loaded with the right parameters for a Shimano Nexus 8-speed. If you have a 7-speed: you'll have to find the right settings for the servo pulsewidth using the 14.h method. If you have another brand of hubgear, make sure you know what the gear number is that is used to calibrate the hub - refer to your manufacturers manual - and follow the 14.h method.
h. For derailleur gears (each model has its own derailleur travel vs. cable travel relation), you have to write down the pulsewidth you can observe in the Serial Monitor. Make sure you still have the Calibrate script uploaded. When you flip the `TestSwitch` (start at minimum position, then go up), then flip it step by step until your derailleur makes the chain go in the next gear. It is does, write down the pulsewidth. Do this for every gear, then proceed to the last step.

15. Upload the Shift4Me script

a. Run the appropriate script for your gearing system (“hubgear” or “derailleur”). It’s a good idea to immediately save it as a new file (Save as…->choose a new filename) because the script itself will be containing the settings that are unique to your bicycle.
b. With the script opened, you see four tabs in the script window. Go to the second one called “GearData”. It now should look like pic 15.
c. There are two things you need to set here:
    i. The amount of gears your bike has (for example: 9 gears). You have to input this figure in the “GearAmount” parameter, see the top red arrow in pic 15..
    ii. The pulsewidth figures (bottom red arrow in pic 15) you have obtained in step 14.h. Put these in ascending order (for derailleurs) or descending order (for hubgears) in the gear parameters. Make sure you adjust the `index value` (see green arrow in pic 15) as you need to add or remove lines to match your amount of gears. If this is your first encounter with scripts: be careful! If you get one digit, semicolon or something else wrong the script will not work! If you’re new to this, make a copy of the original script so you can always start over easily.
    iii. The gear parameters start counting at zero… So for example if you have a 6-gear bike, your GearAmount will be 6 and the parameters will count from Gears[0] to Gears[5].
d. When all pulsewidth parameters, and the GearAmount, are set: upload the script to the `Arduino`.
e. Now put the bike on a stand (if it’s not still there) and test the Shift4Me: plug in the `TestSwitch`, but don’t plug in the `Sensor`. Turn the power on using the `KeySwitch`. Rotate the pedals, and briefly flip the `TestSwitch` to one side. Your bike gears should move to another gear. Turn the `TestSwitch` to the other side, guess what? It’s returning to its former gear. If this is all going well, you just have to unplug the `TestSwitch`, plug in the `Sensor`, and screw the top onto the `BigBox`. Congrats, you are now the proud owner of an automatic-gearing bike!

Appendix

Components for the standard Shift4Me system.

Below is the list of all you need. There's a file in the Forum's Downloads "components" too. The only thing you'll have to work out locally is the printing of the 3D-printed parts, be sure to check the file called "xxx" in the Downloads section if you contact your local 3D printservice, or are going to print the parts on your own printer.

Bike parts

- Derailleur cable with hose. Length depends on where you want to install the `BigBox`, but xxx cm is sufficiant for the standard setup (on the frame, the vertical saddle bar)
- 2 hose ends
- 1 derailleur cable adjuster
- A sensor (the same type as used for simple speed sensors of bike computers)
- A (strong) ring magnet that you can mount to the magnet holder

Electronics & related

- Arduino Nano board
- Servo (we recommend the PowerHD 1501MG)
- Toggle switch with function (on)off(on). This is just for testing and setup. You can replace this with two push-buttons if you like.
- Two PCB boards with "island contacts".
    - For the main circuit you need a 15x20 hole board
    - For the control unit you need a 5x14 hole board
- 2 resistors 1/4W: 470 ohm
- 1 resistor 1/4W: 220 ohm
- A push-button with built-in LED (Pic 18)
- 1 mini-buzzer 5V
- About 50 cm of **solid** wire to solder the contact lines on the PCB's (small diameter is OK)
- About 20 cm of **flexible** wire to connect the plugs and switches. Best is to use the coloured flatcable type (6-pole), diameter 0.75mm or even smaller. You need a larger diameter for the strong-current connections (1mm minimum)
- Some shrink-tube or isolation tape
- Two battery packs 2200mAh size AA, we recommend Ni-Mh:
    - 5-cell (6V)
    - 2-cell (2.4V)
- Battery charger (Ni-Mh) capable of charging 7-cell battery packs (8.4V), preferably with a discharge function.

- A 4-pole cable that connects the `control unit` to the `BigBox`. For a standard installation a length of 180 cm should do it. Telephone cable is just fine.
- A 3-pole servo connector. You can use “row pin” connectors for these as well.
- Plugs:
  - Mono jack 3.5mm: 1 male plug to connect to the reed sensor, 1 female chassis type (Pic 19)
  - Stereo jack 3.5mm: 1 male plug to connect to the test toggle switch, one female PCB-mounting type
  - Charger plug: chassis type 5.5x2.5 (recommended) (Pic 20)
  - Double pole contact switch “`Key Switch`” suitable for minimum 3A@9V (Pic 21)
  - DIN 4-pole: 1 female plug to connect to the wire coming from the `control unit`, 1 male chassis (Pic 22)
  - *4 screw terminals*
- Tie-wraps: two strong ones (4mm, these will secure the 500gr weighing BigBox) and approx. 4 small ones
- 3 rubber bands, the kind that are being used to mount bike computers and such. You need 2 for the Control Unit and 1 for the Sensor.

Board specifications (Arduino or similar)

- Digital inputs: 3
- Digital outputs: 3 (1 with PWM output)
- Input voltage: 9V

Servo specifications (recommended)

- Torque: >16kg/cm
- Operating voltage: 6V
- Rotation >140°