Function GET_MW – single-thread version:

```
res = call_external(libname, 'GET_MW', Lparms, Rparms, Parms, $
                    T_arr, DEM_arr, DDM_arr, RL)
```

Function parameters:

0. Lparms – 5-element long integer array of dimensions and global (for all voxels) integer parameters (see below).
1. Rparms – 3-element double array of global (for all voxels) real parameters (see below).
2. Parms – array of LOS parameters, 15 × Nz elements, double (see below).
3. T_arr – array of temperatures where DEM/DDM are specified, NT elements, double, in K (the temperature grid is assumed to be the same in all voxels, and the same for both DEM and DDM).
4. DEM_arr – array of DEM, NT × Nz, double, in cm^{-6} K^{-1}.
5. DDM_arr – array of DDM, NT × Nz, double, in cm^{-3} K^{-1}.
6. RL – input/output array, 7 × Nf, double:
   c. first row (RL[0, *]) – emission frequencies, in GHz;
   d. other rows – emission intensities, in sfu.

Array of dimensions and global integer parameters:

Lparms = [Nz, Nf, NT, DEM_key, DDM_key]

0. Nz – number of voxels along LOS;
1. Nf – number of frequencies in the spectrum;
2. NT – number of temperatures in the T_arr array; must be ≥ 2 – otherwise DEM/DEM are ignored;
3. DEM_key – global DEM on/off key.
   a. 0: DEM is enabled: it can be used in all or some voxels, depending on the local DEM on/off keys (see below).
   b. ≠ 0: DEM is disabled for all voxels, regardless of the local DEM on/off keys.
4. DDM_key – global DDM on/off key: same as above, but for DDM.

Array of global real parameters:

Rparms = [S, f_0, Δf]

0. S – visible source area, in cm^2.
1. f_0 – starting frequency of the spectrum, in Hz:
   a. is used, only if $f_0 > 0$;
   b. if $f_0 \le 0$, the frequencies are taken from the RL[0, *] array.
2. Δf – logarithmic frequency step (is used only if $f_0 > 0$).

Array of parameters Parms (for a single voxel):

0. Parms[0] = Δz – voxel length, in cm.
1. Parms[1] = T_0 – plasma temperature, in K (is used if DEM or DDM are not specified).
2. Parms[2] = n_0 – either electron concentration or total atomic concentration (depending on other parameters), in cm^{-3} (is used if DEM or DDM are not specified).
3. Parms[3] = B – magnetic field strength, in G.
4. Parms[4] = θ – viewing angle, in degrees.
5. Parms[5] = φ – magnetic field azimuthal angle, in degrees.
6. Parms[6] – emission mechanism flag (rounded to the nearest integer):
    a. 0: all emission mechanisms (gyroresonance + free-free + contribution of neutrals) are included;
    b. 1: gyroresonance is off;
    c. 2: free-free is off;
    d. 4: contribution of neutrals is off.

    Several flags can be combined by usual or bitwise summation: e.g., mechanism flag = 2 + 4 turns off both free-free and contribution of neutrals, etc.
7. Parms[7] = s_{max} – maximum cyclotron harmonic number.
8. Parms[8] = n_p – proton concentration, in cm^{-3} (is used if DEM or DDM are not specified, and the temperature is low).
9. Parms[9] = n_{HI} – neutral hydrogen concentration, in cm^{-3}.
10. Parms[10] = n_{HeI} – neutral helium concentration, in cm^{-3}.
11. Parms[11] – local DEM on/off key:
    e. 0: DEM is used (provided that NT ≥ 2 and DEM is enabled by the global key);
    f. ≠ 0: DEM in this voxel is ignored even if it is specified; T_0 and n_0 are used instead.
12. Parms[12] – local DDM on/off key: same as above, but for DDM.
13. Parms[13] – element abundance model:
    g. 0: coronal;
    h. 1: photospheric (Caffau);
    i. 2: photospheric (Scott).
14. Parms[14] = Vox_ID – voxel type (coronal / chromospheric / etc.), currently ignored.

Function FindIonizations:

```
a = call_external(libname, 'FindIonizations', n0, T0, n_e, n_H, n_He,  /d_value)
```

Input parameters (double scalar variables only): total atomic (H + He) density n0 [cm^{-3}], plasma temperature T0 [K].

Output parameters (double scalar variables only): concentrations of electrons (n_e), neutral hydrogen (n_H) and neutral helium (n_He), all in cm^{-3}.

Return value: n_e/n0.